DOE/PC/94208--T8

SEVENTH TECHNICAL PROGRESS REPORT

ON

HYDRODYNAMIC MODELS FOR SLURRY

BUBBLE COLUMN REACTORS

JANUARY - MARCH 1996

APRIL 1996

RECEIVED

OCT 28 1996

OSTI

U.S. DEPARTMENT OF ENERGY GRANT

DE-FG22-94PC4208

RECEIVED
USD OE/PETC
36 APR 30 PM 2:35
ACQUISITION & ASSISTANCE DIV.

DIMITRI GIDASPOW

PRINCIPAL INVESTIGATOR

DEPARTMENT OF CHEMICAL AND ENVIRONMENTAL ENGINEERING

ILLINOIS INSTITUTE OF TECHNOLOGY

CHICAGO, ILLINOIS 60616

MASTER

HH
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

U.S. DOE PATENT CLEARANCE NOT REQUIRED

PRIOR TO PUBLICATION OF THIS REPORT

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

HYDRODYNAMIC MODELS FOR SLURRY BUBBLE COLUMN REACTORS

ABSTRACT

The objective of this investigation is to convert our "learning gas-solid-liquid" fluidization model into a predictive design model. The IIT hydrodynamic model computes the phase velocities and the volume fractions of gas, liquid and particulate phase. Model verification involves a comparison of these computed velocities and volume fractions to experimental values.

This report presents the Ph.D. thesis of Mr. Y. Wu. All but a small fraction of his thesis dealt with the work of this project. A summary of his results are as follows.

A hydrodynamic model for multiphase flows, based on the principles of mass conservation, momentum balance and energy conservation for each phase, was developed and applied to model gas-liquid, gas-liquid-solid fluidization and gas-solid-solid separation. To simulate the industrial slurry bubble column reactors, a computer program based on the hydrodynamic model was written with modules for chemical reactions (e.g. the synthesis of methanol), phase changes and heat exchangers. Also the kinetic theory was programmed to compute the viscosity of the solid phase.

In the simulations of gas-liquid two phases flow system, the gas hold-ups, computed with a variety of operating conditions such as temperature, pressure, gas and liquid velocities, agree well with the measurements obtained at Air Products' pilot plant. The hydrodynamic model has more flexible features than the previous empirical correlations in predicting the gas hold-up of gas-liquid two-phase flow systems.

In the simulations of gas-liquid-solid bubble column reactors with and without slurry circulation, the code computes volume fractions, temperatures and velocity distributions for the gas, the liquid and the solid phases, as well as concentration distributions for the species (CO , H_2 , CH_3OH , ...), after startup from a certain initial state. A kinetic theory approach is used to compute a solid viscosity due to particle collisions. Solid motion and gas-liquid-solid mixing are observed on a color PCSHOW movie made from computed time series data. The steady state and time average catalyst concentration profiles, the slurry height and the rates of methanol production agree well with the measurements obtained at an Air Products' pilot plant. The temperature movie shows heat transfer behavior of the internal heat exchangers. The simulations may be useful as a guide for industrial reactor design such as for determining reactor size, distributor design, positions of heat exchangers, mixing type, and erosion of heat exchanger tubes.

SIMULATION OF METHANOL SYNTHESIS
IN SLURRY BUBBLE COLUMN REACTORS

BY
YUANXIANG WU

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved

Adviser

Chicago, Illinois
May 1996

ACKNOWLEDGMENT

First, I would like to express my sincere appreciation and gratitude to my thesis advisor, Professor Dimitri Gidaspow, for his helpful advice and guidance during the course of this work.

Sincere thanks are due to my colleagues, Ms. Bing Sun, Professor Huilin Lu, Dr. Aubrey L. Miller, Dr. Isaac K. Gamwo, Dr. Apichai Therdthianwon, Dr. Mitra Bahary, Ms. Diana Matonis, Mr. Reza Mostafi and Mr. Augusto Neri, for their help and cooperation.

Furthermore, my thanks are extended to the staff of the Department of Chemical and Environmental Engineering in IIT for their support.

Finally and most specially, I would like to thank my wife Hongli for her invaluable support and encouragement during the pursuit of my doctorate.

Financial support for this study was provided by the US Department of Energy Grant No. DE-FG22-94PC94208.

Y. W.

TABLE OF CONTENTS

ACKNOWLEDGMENT	Page iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF SYMBOLS	xii
ABSTRACT	xv
CHAPTER	
I. INTRODUCTION	1.1
II. DESCRIPTIONS OF HYDRODYNAMIC MODELS	2.1
2.1 Introduction	2.1
2.2 Hydrodynamic Models	2.1
2.3 Constitutive Equations	2.3
2.4 Solid Viscosity from Kinetic Theory Approach	2.6
2.5 Reaction and Mass Transfer	2.8
2.6 Species Balances	2.8
III. GAS HOLDUP OF GAS-LIQUID FLUIDIZATION	3.1
3.1 Introduction	3.1
3.2 Hydrodynamic Models	3.2
3.3 Operating Conditions	3.4
3.4 Numerical Considerations	3.5
3.5 Results and Discussions	3.8
IV. METHANOL SYNTHESIS WITH SLURRY CIRCULATION	4.1
4.1 Introduction	4.1
4.2 Reaction Kinetics and Mass Transfer	4.1
4.3 Species Balance	4.3
4.4 Mathematical Models (Empirical)	4.3
4.5 Operation Conditions	4.5
4.6 Numerical Considerations	4.6
4.7 Computational Results and Discussions	4.9

CHAPTER	Page
V. METHANOL SYNTHESIS WITHOUT SLURRY CIRCULATION	5.1
5.1 Introduction	5.1
5.2 Reaction and Kinetic Model	5.1
5.3 Mathematics Models (Kinetic Theory)	5.2
5.4 Operation Conditions	5.6
5.5 Numerical Considerations	5.8
5.6 Computational Results and Discussions	5.11
5.7 Convergence Check	5.20
 VI. SEPARATION OF PYRITE FROM COAL IN AN INCLINED ELECTRO-FLUIDIZED BED	 6.1
6.1 Introduction	6.1
6.2 Hydrodynamic Model	6.4
6.3 Operation Conditions	6.7
6.4 Numerical Considerations	6.7
6.5 Computational Results and Discussions	6.12
 VII. SUMMARY AND RECOMMENDATIONS	 7.1
 APPENDIX	
A. NUMERICAL BASIS OF PROGRAM MFREK	A.1
B. ORGANIZATION AND USER MENU OF MFREK	B.1
C. SOURCE FORTRAN CODE OF MFREK AND A SAMPLE INPUT DATA	C.1
BIBLIOGRAPHY	D.1

LIST OF TABLES

Table	Page
3.1 Computed and Measured Gas Holdups at Reactor Height 137cm	3.6
4.1 Material Balance: (CO:H ₂ =1:3)	4.9
4.2 Comparison of Simulation and Air Products' (1987) RUN E-2-B	4.10
5.1 Material Balance: (CO:H ₂ =1.5:1)	5.12
5.2 Comparison of Simulation and Air Products' (1991) RUN E-8.1	5.12
6.1 Case 1: Dense Operation with High Pyrite Content	6.9
6.2 Case 2: Dense Operation with Low Pyrite Content	6.9
6.3 Case 3: Pneumatic Operation	6.9

LIST OF FIGURES

Figure	Page
3.1 Reactor Operation and Simulation Grid	3.7
3.2 Computed and Experimental Gas Holdup Profiles	3.9
3.3 Computed and Experimental Axial Gas Holdup Profiles (Effect of Liquid Velocity)	3.11
3.4 Computed and Experimental Gas Holdup vs Gas Velocity (at Reactor Height=137cm)	3.12
3.5 Effect of Temperature on Gas Holdup (at Reactor Height=137cm)	3.13
3.6 Effect of Pressure on Gas Holdup (at Reactor Height=137cm)	3.14
3.7 Comparison of Experimental and Computed Gas Holdup	3.15
4.1 Reactor Operation with Slurry Circulation and Simulation Grid	4.8
4.2 Computed and Experimental Axial Catalyst Volume Fraction Profiles	4.11
4.3 Computed Gas, Liquid and Solid Axial Volume Fraction Profiles	4.12
4.4 Computed Axial Methanol Concentration Profile	4.13
4.5a The Catalyst Concentration Profiles as a Function of Time	4.14
4.5b The Catalyst Concentration Profiles as a Function of Time	4.15
4.5c The Catalyst Concentration Profiles as a Function of Time	4.16
4.6a The Methanol Concentration Profiles as a Function of Time	4.17
4.6b The Methanol Concentration Profiles as a Function of Time	4.18
4.6c The Methanol Concentration Profiles as a Function of Time	4.19
5.1 Reactor Operation without Slurry Circulation and Simulation Grid	5.7
5.2 Gas Mass Flowrate at Reactor Top	5.13

Figure	Page
5.3 Time average G-L-S Volume Fraction Profiles	5.14
5.4 Methanol Concentration Profiles from IIT's Hydrodynamic, One-dimensional and Vikings' Models	5.15
5.5a The Gas Volume Fraction Profiles as a Function of Time	5.16
5.5b The Gas Volume Fraction Profiles as a Function of Time	5.17
5.5c The Gas Volume Fraction Profiles as a Function of Time	5.18
5.5d The Gas Volume Fraction Profiles as a Function of Time	5.19
5.6a The Solid Volume Fraction Profiles as a Function of Time	5.21
5.6b The Solid Volume Fraction Profiles as a Function of Time	5.22
5.6c The Solid Volume Fraction Profiles as a Function of Time	5.23
5.6d The Solid Volume Fraction Profiles as a Function of Time	5.24
5.7a The Methanol Concentration Profiles as a Function of Time	5.25
5.7b The Methanol Concentration Profiles as a Function of Time	5.26
5.7c The Methanol Concentration Profiles as a Function of Time	5.27
5.7d The Methanol Concentration Profiles as a Function of Time	5.28
5.7e The Methanol Concentration Profiles as a Function of Time	5.29
5.8a The Temperature Profiles as a Function of Time	5.30
5.8b The Temperature Profiles as a Function of Time	5.31
5.8c The Temperature Profiles as a Function of Time	5.32
5.9 Instant Rate of Reaction (gmol/s/cm^3)	5.33
5.10 Solid Viscosity Distribution	5.34
5.11 Solid Granular Temperature	5.35
5.12 Gas Transient Flow Pattern	5.36

Figure	Page
5.13 Time Series of Gas Flow Patterns	5.37
5.14 Time Average G-L-S Volume Fraction Profiles in Different Grid	5.39
5.15 Gas Flowrate at Reactor Top and Frequence Response (Fine Grid)	5.40
5.16 Gas Flowrate at Reactor Top and Frequence Response (Coarse Grid)	5.41
5.17a Gas Flow Patterns (Fine Grid)	5.42
5.17b Gas Flow Patterns (Fine Grid)	5.43
5.18a Gas Flow Patterns (Coarse Grid)	5.44
5.18b Gas Flow Patterns (Coarse Grid)	5.45
6.1 Simulation Grid of an Inclined Electro-Fluidized Bed	6.8
6.2 Coal Recovery vs Pyrite Content of Coal	6.13
6.3a Pyrite and Coal Volume Fraction Profiles and Flow 48 Patterns. Dense Case: Air/Coal (21.6% Pyrite) = 1/1	6.14
6.3b Pyrite and Coal Volume Fraction Profiles and Flow Patterns. Dense Case: Air/Coal (2.5% Pyrite) = 1/1	6.15
6.3c Pyrite and Coal Volume Fraction Profiles and Flow Patterns. Dilute Case: Air/Coal (1.7% Pyrite) = 1/0.012	6.16
6.4a The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.18
6.4b The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.19
6.4c The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.20
6.4d The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.21
6.4e The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.22
6.4f The Coal Volume Fraction Profiles as a Function of Time (Case 2)	6.23
6.5a The Pyrite Volume Fraction Profiles as a Function of Time (Case 2)	6.24
6.5b The Pyrite Volume Fraction Profiles as a Function of Time (Case 2)	6.25

Figure	Page
6.5c The Pyrite Volume Fraction Profiles as a Function of Time (Case 2) . . .	6.26
6.5d The Pyrite Volume Fraction Profiles as a Function of Time (Case 2) . . .	6.27
6.5e The Pyrite Volume Fraction Profiles as a Function of Time (Case 2) . . .	6.28
6.5f The Pyrite Volume Fraction Profiles as a Function of Time (Case 2) . . .	6.29
6.6a The Coal Flow Patterns as a Function of Time (Case 2)	6.30
6.6b The Coal Flow Patterns as a Function of Time (Case 2)	6.31
6.6c The Coal Flow Patterns as a Function of Time (Case 2)	6.32
6.6d The Coal Flow Patterns as a Function of Time (Case 2)	6.33
6.6e The Coal Flow Patterns as a Function of Time (Case 2)	6.34
6.6f The Coal Flow Patterns as a Function of Time (Case 2)	6.35
6.7a The Pyrite Flow Patterns as a Function of Time (Case 2)	6.36
6.7b The Pyrite Flow Patterns as a Function of Time (Case 2)	6.37
6.7c The Pyrite Flow Patterns as a Function of Time (Case 2)	6.38
6.7d The Pyrite Flow Patterns as a Function of Time (Case 2)	6.39
6.7e The Pyrite Flow Patterns as a Function of Time (Case 2)	6.40
6.7f The Pyrite Flow Patterns as a Function of Time (Case 2)	6.41
6.8a The Coal Flow Patterns as a Function of Time (Case 1)	6.42
6.8b The Coal Flow Patterns as a Function of Time (Case 1)	6.43
6.8c The Coal Flow Patterns as a Function of Time (Case 1)	6.44
6.8d The Coal Flow Patterns as a Function of Time (Case 1)	6.45
6.9a The Pyrite Flow Patterns as a Function of Time (Case 1)	6.46
6.9b The Pyrite Flow Patterns as a Function of Time (Case 1)	6.47

Figure	Page
6.9c The Pyrite Flow Patterns as a Function of Time (Case 1)	6.48
6.9d The Pyrite Flow Patterns as a Function of Time (Case 1)	6.49
6.10a The Coal Flow Patterns as a Function of Time (Case 3)	6.50
6.10b The Coal Flow Patterns as a Function of Time (Case 3)	6.51
6.10c The Coal Flow Patterns as a Function of Time (Case 3)	6.52
6.10d The Coal Flow Patterns as a Function of Time (Case 3)	6.53
6.10e The Coal Flow Patterns as a Function of Time (Case 3)	6.54
6.11a The Pyrite Flow Patterns as a Function of Time (Case 3)	6.55
6.11b The Pyrite Flow Patterns as a Function of Time (Case 3)	6.56
6.11c The Pyrite Flow Patterns as a Function of Time (Case 3)	6.57
6.11d The Pyrite Flow Patterns as a Function of Time (Case 3)	6.58
6.11e The Pyrite Flow Patterns as a Function of Time (Case 3)	6.59
6.12 The Coal Viscosity vs Concentration (Dense Case)	6.61
6.13 The Pyrite Viscosity vs Concentration (Dense Case)	6.62
6.14 The Pyrite Viscosity vs Concentration (Dilute Case)	6.63
6.15 Comparison of Coal and Pyrite viscosities (Dense Case)	6.64
A.1 The Computational Mesh	A.9
A.2 The Staggered Computational Mesh for Momentum equations	A.11
A.3 The Computational Sweep	A.19
A.4 The Secant method for Pressure Iteration	A.22
B.1 The Program Flow Chart	B.3

LIST OF SYMBOLS

Symbol	Definition
A	coefficient matrix
a	interfacial area per unit volume (cm^2/cm^3)
B	coefficient matrix
C_D	drag coefficient
C_j^l	concentration of j in bulk liquid phase
C_j^{g-l}	concentration of j in $g-l$ interface
C_k	fluctuating velocity of particle
C_{pg}	specific heat of gas
C_{pk}	specific heat of phase k
d_k	diameter of solid particle or liquid droplet
e_k	restitution coefficient of phase k
F_g	forces acting on gas phase
F_k	forces acting on phase k
E	strength of electric field
f_j	fugacity of j
H_j	Henry's constant of j
H_g	enthalpy of gas
H_k	enthalpy of phase k
h_{vk}	gas-phase k heat transfer coefficient
$G(\cdot)$	solid compressive stress modules
g	gravity acceleration
g_{0k}	radial distribution of phase k
$[I]$	unit tensor
K_{eq}	reaction equilibrium constant
K_r	reaction kinetic coefficient
K_{j^l}	mass transfer coefficient of j in liquid phase
k_g	thermal conductivity of gas

Symbol	Definition
k_k	thermal conductivity of phase k
k_g^0	mean thermal conductivity of gas
k_k^0	mean thermal conductivity of phase k
k_k^*	effective thermal conductivity of phase k
M^j	molecular weight of j
$\overline{M}_g$	average gas molecular weight
M_k	number of reactions in phase k
$\dot{m}_k$	rate of generation of phase k
N	total number species
Nu_k	Nusselt number
P_g	gas pressure
P_k	pressure of phase k
Pr	Prandtl number
q_k	surface charge of phase k
r_{ik}	rate of ith reaction in phase k
R	gas constant
R_k^j	rate of mass transfer, j specie in phase k
Re_k	Reynolds number
T	temperature
T_g	gas temperature
T_k	temperature of phase k
T_g^0	reference gas temperature
T_k^0	reference temperature of phase k
u	velocity in x direction
v	velocity in y direction
y_k^j	weight fraction of i in phase k
z	compressible factor

Greek letters:

α_{ik}^j	stoichiometric coefficient of ith reaction in k phase ξ_k
-----------------	---

Symbol	Definition
β_{kl}	frictional coefficient between phase k and l
γ_k	collisional energy dissipation
ΔH_{ik}	reaction heat of ith reaction in phase k
ε_k	volume fraction of phase k
Θ	granular temperature
κ_k	conductivity of fluctuating energy
μ_k	viscosity of phase k
ξ_k	bulk viscosity
ρ_k	density of phase k
τ_{ck}	cohesive force
τ_k	shear stress of phase k
Φ_k	energy dissipation
Ψ_k	sphericity of particle or droplet

Subscripts:

g, ℓ, s	gas, liquid, solid respectively
i	ith reaction
k	solid or liquid phase

Superscripts

j	species
-----	---------

ABSTRACT

In this study, a hydrodynamic model for multiphase flows, based on the principles of mass conservation, momentum balance and energy conservation for each phase, was developed and applied to model gas-liquid, gas-liquid-solid fluidization and gas-solid-solid separation. To simulate the industrial slurry bubble column reactors, a computer program based on the hydrodynamic model was written with modules for chemical reactions (e.g. the synthesis of methanol), phase changes and heat exchangers. Also the kinetic theory was programmed to compute the viscosity of the solid phase.

The slurry bubble column reactor, a typical multiphase flow system, is used to produce methanol from syn-gas (CO , H_2 , ...). The hydrodynamic model was used to simulate the reactor's time-depended dynamic behavior. The code computes volume fractions, temperatures and velocity distributions for the gas, the liquid and the solid phases, as well as concentration distributions for the species (CO , H_2 , CH_3OH , ...), after startup from a certain initial state. In this simulation, the gas is treated as a continuous phase while the solid (catalyst) and the liquid (wax) as a dispersed phase. A kinetic theory approach is used to compute a solid viscosity due to particle collisions. Solid motion and gas-liquid-solid mixing are observed on a color PCSHOW movie made from computed time series data. The steady state and time average catalyst concentration profiles and the rate of methanol production agree well with the measurements obtained at an Air Products' pilot plant. The temperature movie shows heat transfer behavior of the internal heat exchangers. The simulations may be useful as a guide for industrial reactor design

such as for determining reactor size, distributor design, positions of heat exchangers, mixing type, and erosion of heat exchanger tubes.

The model was also applied to understand the hydrodynamics of electrostatic separation of pyrites from coal. The model was modified to include an external electric force. The kinetic theory has been applied to predict the viscosities for both solid phases, coal and pyrites. The study showed that pyrites can be removed from coal in an inclined electro-fluidized bed. A series of pictures from simulations show the hydrodynamic behavior, such as solid flow patterns and pyrites concentrations, in the electrostatic separator. This study indicates that dilute feeding results a high coal recovery and leads to an improved design of electrostatic separators.

CHAPTER I

INTRODUCTION

For the foreseeable future, liquid hydrocarbon fuels will play a significant role in the transportation sector of the world. Factors favoring these fuels include convenience, high energy density, and the vast existing infrastructure for their production and use. A scenario and time frame are developed in which declining world resources will generate a shortage in petroleum supply that can be alleviated in part by utilizing the abundant coal resource base. One option is direct or indirect coal conversion to liquid transportation fuels.

Coal can be converted into liquid fuels by liquefaction process. One important route for indirect liquefaction is hydrogenation of carbon monoxide which yield methanol at high yields. Methanol is suitable for direct use as a turbine and transportation fuel, although its energy content on a volume basis is only half that of gasoline. It can also be used as a feedstock to produce high octane gasoline.

For methanol production technologies, considerable research and development still need to be undertaken. To produce liquid phase methanol products, several reactor systems such as fixed bed reactor, entrained bed reactor, fixed fluidized bed reactor and slurry bubble column reactor, have been developed. Among these reactor systems, the three-phase slurry bubble column reactors provide an attractive alternative to the others since they can be operated with low H_2/CO ratio (<0.6) synthesis gas.

The Air Products and Chemicals (1987) had developed a methanol synthesis technology with the slurry bubble column reactor at its LaPorte Pilot Plant since the early

80's. The liquid phase methanol was obtained during a number of test runs with variety conditions.

To test high catalyst concentration using a commercial synthesis catalyst with CO-rich gas and to evaluate the short-term activity of a commercial catalyst with CO-rich gas, a series of runs was conducted (Air Products 1987). These runs suggested that the hydrodynamics of the slurry bubble reactor with high-catalyst slurry concentration needs to be studied further.

The objective of this study is to improve the hydrodynamics understanding on the slurry bubble column reactors using IIT's hydrodynamic models and computer code.

The IIT's hydrodynamic code, original developed from the K-FIX program (Rivard and Torrey, 1977), was able to predict a great deal of the behaviors of bubbling beds (Gidaspow and Ettehadieh 1983a, Gidaspow et al. 1983b, Ettehadieh 1982, Syamlal 1985, and Gidaspow et al. 1985 and Gamwo 1992), circulating fluidized bed (Therdthianwong 1994 and Ding 1990) and gas-liquid-solid three-phase fluidized bed (Jayaswal 1991 and Bahary 1994). Gidaspow's book (Gidaspow 1994), "Multiphase Flow and Fluidization, Continuum and Kinetic Theory Descriptions," detail describes these fundamental approaches of IIT's hydrodynamic models. His kinetic theory had been programmed to the code to compute the particulate viscosities and solids pressure (Ding and Gidaspow 1990; Gidaspow et al. 1991).

To simulate the slurry bubble column reactors, the IIT's code has been modified properly to satisfied particular needs in this study. For instance, modules for chemical reactions (particular for the synthesis of methanol), phase changes and heat exchangers were added to the code. Also the kinetic theory was used to compute the viscosity of the

catalyst. The models and governing equations of the hydrodynamics, chemical kinetics, heat and mass transfers are described in Chapter II.

To investigate the effects of gas and liquid superficial velocities, operation temperature and pressure on the gas hold-up, a number of simulation runs for gas hold-up were made. Chapter III describes these gas hold-up runs. The gas hold-ups are very close to those measurements obtained from the Air Products' (1987) pilot plant.

Air Products' (1987) RUN E-2 series were the operations of liquid phase methanol with the external slurry circulation. In this study, a dynamic simulation for the RUN E-2-B was conducted. As described in Chapter IV, the computed time-and-cross-section-area-averaged catalyst concentration profiles match well with Air Products' NDG (Nuclear Density Gauge) measurements.

Air Products' (1991) RUN E-8 series were the operations of liquid phase methanol without the external slurry circulation. In this study, a dynamic simulation for the RUN E-8.1 was conducted. The kinetic theory had been used to compute the catalyst viscosities and energy equation had been solved to predict the temperature profiles in count of the internal heat exchangers. As described in Chapter V, the slurry height, methanol production and gas-liquid-solid volume fraction profiles agreed well with that obtained at the Air Products' (1991) pilot plant. A simulation with a finer grid was carried out to investigate the effect of grid on the accuracy and convergence of the numerical computation.

As part of this thesis research, a series of simulations for the coal-pyrite separation in an inclined electro-fluidized bed was performed. Chapter VI describes the results of these simulations. The kinetic theory had been applied to predict the viscosities for both

solids, coal and pyrite. The study had achieved that the pyrite can be removed in a great deal from coal in an inclined electro-fluidized bed.

In Appendix A, a description of governing hydrodynamic equations, the finite difference equations and the solution techniques are presented. Appendix B describes a detail user menu for the multiple functional code, MFREK, which stands Multiphase Flow with Reactions, Energy equations and Kinetic theory. Appendix C lists the source FORTRAN code of MFREK.

CHAPTER II

DESCRIPTIONS OF HYDRODYNAMIC MODELS

2.1 Introduction

The hydrodynamic approach to multiphase flow systems is based on the principles of mass conservation, momentum balance and energy conservation for each phase. This approach is similar to that of Soo (1967) for multiphase flow and of Jackson (1985) for fluidization. Gidaspow (1986), recently reviewed such models as they apply to fluidization and analyzed the necessary steps required to ensure that momentum equations satisfy the Darcy's law and Archimedes' principles.

A transient, two-dimensional hydrodynamic models for multiphase flow of fluid and particulate phases based on the generalization of Navier-Stokes equations was developed at IIT. The review by Gidaspow (1986) showed that inviscid two-fluid models were able to predict a great deal of the behavior of bubbling beds because the dominant mechanism of energy dissipation is the drag between the particles and the fluid. Ding and Gidaspow (1990) extended the model to include solids viscosity from kinetic theory of granular flows and used it to predict the behavior of bubbling beds. The present model has been extended the kinetic theory to gas-liquid-solid fluidized beds and gas-solid-solid binary systems. Besides, a general model for chemical and mass transfer between phases has been developed.

2.2 Hydrodynamic Models

The hydrodynamic models for gas-liquid-solid or gas-solid-solid binary system with phase changes are described as followings.

2.2.1 Continuity Equations. The conservation of mass can be interpreted as that:

rate of mass accumulation + convection due to flow = production due to mass transfer.

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = \dot{m}_g$$

Liquid and Solid phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k) = \dot{m}_k$$

In above equations, the volume fractions of each phase must be added up to unity, and they can be related by

$$\epsilon_g + \epsilon_\ell + \epsilon_s = 1$$

2.2.2 Momentum Equations. The momentum equation for each phase can be interpreted as Newton's first law:

mass x acceleration = sum of forces acting on the phase.

As described in Chapter 2 of the book by Gidaspow (1994), there are three different types of momentum balances, model A, model B and model C. In model A, the pressure drop occurs in each involving phase. An alternative way to treat the pressure drop is to assume that it occurs only in the gas phase or the continuous phase. This is called as model B (Gidaspow 1977, Arastoopour and Gidaspow 1979). Model C is similar to model B except of the solid velocity which is replaced with the relative velocity ($\mathbf{v}_s - \mathbf{v}_g$). Lyczkowski et al. (1978) had found that the momentum equations with model B is well-posed and possesses all real characteristics.

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g v_g) + \nabla \cdot (\epsilon_g \rho_g v_g v_g) = \epsilon_g \rho_g F_g + \sum_{m=\ell, s} \beta_{gm} (v_m - v_g) + \nabla \cdot [\tau_g] + \dot{m}_g v_g$$

Liquid and Solid Phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k v_k) + \nabla \cdot (\epsilon_k \rho_k v_k v_k) = \epsilon_k \rho_k F_k + \sum_{m=g, \ell, s} \beta_{km} (v_m - v_k) + \nabla \cdot [\tau_k] + \dot{m}_k v_k$$

2.2.3 Energy Equations. To treat non-isothermal multiphase flow systems, the energy equations are needed to compute the heat transfer due to heat exchangers and chemical reactions. The equations are written based on the principle of thermodynamics first law:

rate of energy accumulation = heat received - work done by the system

Gas Phase:

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_g \rho_g H_g) + \nabla \cdot (\epsilon_g \rho_g H_g v_g) = & \left(\frac{\partial P_g}{\partial t} + P_g \nabla \cdot v_g \right) + \nabla \cdot (k_g \nabla T_g) + \Phi_g + \sum_i r_{ig} \Delta H_{ig} \\ & + \sum_{m=\ell, s} \left\{ h_{vm} (T_m - T_g) + \beta_{gm} (v_m - v_g)^2 - \dot{m}_m H_g \right\} \end{aligned}$$

Liquid and Solid Phases: ($k=\ell, s$)

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_k \rho_k H_k) + \nabla \cdot (\epsilon_k \rho_k H_k v_k) = & h_{vk} (T_g - T_k) + \sum_{m=g, \ell, s} \beta_{km} (v_m - v_k)^2 + \dot{m}_k H_g \\ & + \nabla \cdot (k_k \nabla T_k) + \sum_i r_{ik} \Delta H_{ik} + \Phi_k \end{aligned}$$

2.3 Constitutive Equations

In order to solve the set of conservation equations of mass, momentum and energy, all the constitutive equations must be defined.

2.3.1 Equation of State. The gas phase is compressible fluid and obeys the equation of state.

$$\rho_g = \frac{\overline{M}_g P_g}{zRT_g}$$

2.3.2 Drag Coefficients. The drag coefficients for model B are based on Ergun equation and empirical correction depending on the gas porosity.

$$\beta_{gk} = \beta_{kg} = 150 \frac{(1-\epsilon_g)\epsilon_k \mu_g}{(\epsilon_g d_k \psi_k)^2} + 1.75 \frac{\rho_g \epsilon_k |v_g - v_k|}{\epsilon_g d_k \psi_k} \quad \epsilon_g > 0.8$$

$$\beta_{gk} = \beta_{kg} = \frac{3}{4} C_D \frac{\rho_g \epsilon_k |v_g - v_k|}{d_k \psi_k} \epsilon_g^{-2.65} \quad \epsilon_g < 0.8$$

where

$$C_D = \frac{24}{Re_k} (1 + 0.15 Re_k^{0.687})$$

$$Re_k = \frac{\rho_g \epsilon_g |v_g - v_k| d_k \psi_k}{\mu_g} \quad Re_k = 1000 \quad \text{if} \quad Re_k > 1000$$

$$\beta_{\ell s} = \beta_{s\ell} = \frac{3}{2} (1+e) \frac{\rho_s \rho_\ell \epsilon_s \epsilon_\ell |v_\ell - v_s|}{\rho_s d_s^3 + \rho_\ell d_\ell^3} (d_s + d_\ell)^2$$

2.3.3 External Forces Acting on Each Phase. The external forces acting on each phase consist of the gravity, electric or magnetic force if applied, etc.

$$F_g = \frac{g}{\epsilon_g}$$

$$F_k = \frac{g}{\epsilon_g} \left(1 - \frac{1}{\rho_k} \sum_{m=g,\ell,s} \epsilon_m \rho_m \right) + q_k E \quad k = \ell, s$$

The q_k is the particle surface charge and the E is the strength of electric field applied.

2.3.4 Shear Stresses. The gas is considered as a Newtonian fluid. The solid shear stress consists of the solid press (P_k), cohesive force ($\tau_{\alpha k}$), and shear rate due to solid shear viscosity (μ_k) and solid phase bulk viscosity (ξ_k).

$$[\tau_g] = \left\{ -P_g - \frac{2}{3} \mu_g \epsilon_g \nabla \cdot \mathbf{v}_g \right\} [I] + \mu_g \epsilon_g [\nabla \mathbf{v}_g + (\nabla \mathbf{v}_g)^T]$$

$$[\tau_k] = \left\{ -P_k + \tau_{ck} + \left(\xi_k - \frac{2}{3} \mu_k \right) \nabla \cdot \mathbf{v}_k \right\} [I] + \mu_k [\nabla \mathbf{v}_k + (\nabla \mathbf{v}_k)^T] \quad k = \ell, s$$

The empirical cohesive force can be given:

$$\tau_{ck} = 10^{10.6\epsilon_k + 5.5} \quad k = \ell, s$$

The solid pressure and viscosities expressions from kinetic theory will be described in section 2.4 of this chapter. Here lists some empirical correlations. ($k = \ell, s$)

$$\mu_k = 5\epsilon_k \quad \xi_k = 0$$

$$\nabla P_k = G(\epsilon_k) \nabla \epsilon_k \quad G(\epsilon_k) = 10^{8.76\epsilon_k - 0.27}$$

2.3.5 Enthalpy

$$H_g = C_{p_g} (T_g - T_g^0)$$

$$H_k = C_{p_k} (T_k - T_k^0)$$

2.3.6 Gas-Solid Heat Transfer. (Gunn's Model)

$$Nu_k = \left\{ (2 + 5\epsilon_k^2)(1 + 0.7 Re_k^{0.2} Pr^{1/3}) + \left(\frac{2}{15} + 1.2\epsilon_k^2 \right) Re_k^{0.7} Pr^{1/3} \right\} Sp_k$$

$$Nu_k = \frac{h_{vk} d_k}{k_g^0} \quad Pr = \frac{C_{p_g} \mu_g}{k_g^0} \quad Sp_k = \frac{6\epsilon_k}{d_k}$$

2.3.7 Gas Phase Heat Transfer

$$k_g^0 = 8.65 \times 10^5 \left(\frac{T_g}{1400} \right)^{1.786}$$

$$k_g = (1 - \sqrt{1 - \epsilon_g}) k_g^0$$

2.3.8 Particulate Phase Heat Transfer

$$\frac{k_k}{k_g^0} = \frac{\sqrt{\epsilon_s}}{(1-\epsilon_g)} \left\{ \phi \frac{k_k^*}{k_g^0} + (1-\phi) \frac{k_k^0}{k_g^0} \right\}$$

$$\frac{k_k^0}{k_g^0} = \frac{2}{A_k} \left\{ \frac{B_k \left\{ (k_k^*/k_g^0) - 1 \right\}}{A_k^2 (k_k^*/k_g^0)} \ln \frac{(k_k^*/k_g^0)}{B_k} \frac{B_k - 1}{A_k} \frac{B_k + 1}{2} \right\}$$

where

$$A_k = 1 - \frac{B_k}{(k_k^*/k_g^0)}$$

$$B_k = 1.25 \left(\frac{\epsilon_k}{\epsilon_g} \right)^{10/9} (1 + 3\chi)$$

$$\chi = \sqrt{\frac{(\sum \epsilon_k \rho_k)(\sum \epsilon_k \rho_k / d_k^2)}{(\sum \epsilon_k \rho_k / d_k)^2}} - 1 \quad k=1,s$$

$$k_k^* = 0.3289 \quad \phi = 7.26 \times 10^{-3}$$

2.4 Solid Viscosity from Kinetic Theory Approach

The solid viscosity or solid shear stresses of multiphase flow may not be constant or as a function of the solid concentration in the rapid shear flow regime. Experiments of Savage and Sayed (1984) have shown that the solid stress is proportional to the square of the shear rate in a certain range of shear rate of simple shear flows. To model the solid shear stress, the statistical approach may be used to formulate the constitutive equations for the solid phase. Analogous to those used in the kinetic theory of dense gases (Chapman and Cowling 1970), several models for the flow of granular materials have been developed based on the assumptions that the collisions between the particles provide the principal mechanism for the transport properties, such as momentum and energy, and that

the velocity distribution function of single-particle is Maxwellian (Savage and Jeffery 1981, Jenkins and Savage 1983). Savage (1983) and Jenkins and Savage (1983) have developed the kinetic theory approach by introducing the concept of restitutive coefficient and by using the Boltzmann transport equation for dense gases. Ding and Gidaspow (1990) extended their approach to particle flow for dense phase. In this study, the kinetic theory approach has been applied to gas-liquid-solid three phase fluidization and gas-solid-solid binary systems.

Fluctuating Energy Equation. ($k=\ell, s$)

$$\frac{3}{2} \left[\frac{\partial}{\partial t} (\epsilon_k \rho_k \Theta_k) + \nabla \cdot (\epsilon_k \rho_k \Theta_k \mathbf{v}_k) \right] = \nabla \cdot (\kappa_k \nabla \Theta_k) - \gamma_k - 3\beta_{kg} \Theta_k + \Phi_k$$

Energy Production Rate

$$\Phi_k = [\tau_k] : \nabla \mathbf{v}_k \quad k=g, \ell, s$$

Conductive Coefficient of Fluctuating Energy.

$$\kappa_k = \frac{2\kappa_{kan}}{(1+e_k)g_{0k}} \left\{ 1 + \frac{6}{5}(1+e_k)g_{0k}\epsilon_k \right\}^2 + 2\epsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

Collisional Energy Dissipation.

$$\gamma_k = 3(1-e_k^2)\epsilon_k^2 \rho_k g_{0k} \Theta_k \left\{ \frac{4\sqrt{\Theta_k}}{d_k \sqrt{\pi}} \nabla \cdot \mathbf{v}_k \right\}$$

Solids Pressure.

$$\mathbf{P}_k = \rho_k \epsilon_k \Theta_k \{ 1 + 2(1+e_k)g_{0k}\epsilon_k \}$$

Solids Bulk Viscosity.

$$\xi_k = \frac{4}{3} \epsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

Solids Shear Viscosity.

$$\mu_k = \frac{2\mu_{k_{dl}}}{(1+e_k)g_{0k}} \left\{ 1 + \frac{4}{5}(1+e_k)g_{0k}\epsilon_s \right\}^2 + \frac{4}{5}\epsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

Where

$$g_{0k} = \left\{ 1 - \left(\frac{\epsilon_k}{\epsilon_{k_{max}}} \right)^{1/3} \right\}^{-1} \quad \mu_{k_{dl}} = \frac{5}{96} \rho_k d_k \sqrt{\pi \Theta_k} \quad \kappa_{k_{dl}} = \frac{75}{384} \rho_k d_k \sqrt{\pi \Theta_k}$$

2.5 Reaction and Mass Transfer

General Reactions

$$\sum_j^N \alpha_{ik}^j S^j = 0 \quad i=1,2,\dots,M_k \quad k=g,\ell,s$$

where S^j represents the j th specie and α_{ik}^j represents the stoichiometric coefficient of the j th specie in the i th reaction of phase k .

$$r_{ik} = f_{ik}(P_k, T_k, \epsilon_k, y_k^1, y_k^2, \dots, y_k^N, \dots) \quad i=1,2,\dots,M_k \quad k=g,\ell,s$$

The overall rate r_{ik} may be a function of temperature, pressure, catalyst concentration and compositions.

Mass Transfer Between Phases

$$R_k^j = f_k^j(P_k, T_k, a_k, y_k^1, y_k^2, \dots, y_k^N, P_g, T_g, y_g^1, y_g^2, \dots, y_g^N, \dots) \quad k=\ell,s$$

The mass transfer rate R_k^j of the j th specie between the phase k and the continuous phase (gas) may be a function of temperature, pressure, interfacial area and compositions.

2.6 Species Balances ($j=1, 2, \dots, N$)

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g y_g^j) + \nabla \cdot (\epsilon_g \rho_g y_g^j \mathbf{v}_g) = \sum_{i=1}^{M_i} \alpha_{ig}^j \mathbf{M}^j \mathbf{r}_{ig} - \sum_{k=\ell, s} \mathbf{M}^j \mathbf{R}_k^j$$

Liquid and Solid Phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k y_k^j) + \nabla \cdot (\epsilon_k \rho_k y_k^j \mathbf{v}_k) = \sum_{i=1}^{M_k} \alpha_{ik}^j \mathbf{M}^j \mathbf{r}_{ik} + \mathbf{M}^j \mathbf{R}_k^j$$

$$\sum_j^N y_k^j = 1 \quad k=g, \ell, s$$

Phase Changes: ($k=\ell, s$)

$$\dot{m}_k = \sum_j^N \mathbf{M}^j \mathbf{R}_k^j \quad \dot{m}_g = - \sum_{k=\ell, s} \dot{m}_k$$

CHAPTER III

GAS HOLD-UP OF GAS-LIQUID FLUIDIZATION

3.1 Introduction

The gas holdup in slurry bubble columns is affected by a number of factors which include: physical properties and coalescing behavior of liquid, gas and liquid velocities, column diameter, gas distributor design. The gas holdup also depends on the operating flow regime in the column. The development of a general correction which takes into account the effect of all these variables is indeed a difficult task.

According to the traditional opinion, gas and liquid velocities have most significant effect on gas holdup. Recent studies, however, show that gas holdup in bubble columns generally increases with increasing operating pressure, hence gas density (Idogawa et al. 1987, Clark 1990 and Kojima et al. 1991). The influence of pressure has been found to depend on pressure level, distributor type, gas and liquid velocities. These results show that a correlation for gas holdup should account for the effect of operating pressure (or gas density) too. Most of the literature correlations, however, is based on data obtained at atmospheric pressure and usually for one gas, as a result these equations do not account for any influence of gas density. A few gas holdup equations have been developed based on high pressure operation (Idogawa et al. 1986 and 1987) or experiments using various gases (Reilly et al. 1986 and Hammer et al. 1984). There is, however, a lack of systematic research to investigate the influence of pressure on gas holdup in slurry bubble columns. Correlations for gas holdup in slurry bubble columns have been proposed by a number of researchers including Deckwer et al. (1980), Koide et al. (1984), Sada et al. (1986), Sauer

and Hempel (1987) and Bukur et al. (1990). These experimental correlations can not take into account the effect of pressure, temperature, gas and liquid velocities. In this chapter, hydrodynamic models for gas holdup are developed and used to predict the gas holdup of LaPorte's RUN E-3 series (Air Products 1987).

3.2 Hydrodynamic Models

The hydrodynamic models employed in the simulations of gas-liquid fluidization are based on the principle of conservations of mass and momentum for both gas and liquid phases (Gidaspow 1994). No energy equation is needed because of an isothermal assumption. Gas is treated as continuous phase while liquid as disperse phase. The governing equations are expressed as following:

3.2.1 Conservation of Mass

Gas Phase.

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = 0$$

Liquid Phase.

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

3.2.2 Conservation of Momentum

Gas Phase.

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \mathbf{v}_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g \mathbf{v}_g) = \epsilon_g \rho_g \mathbf{F}_g + \beta_{gk}(\mathbf{v}_k - \mathbf{v}_g) + \nabla \cdot [\boldsymbol{\tau}_g]$$

Liquid Phase.

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k \mathbf{v}_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k \mathbf{v}_k) = \epsilon_k \rho_k \mathbf{F}_k + \beta_{kg}(\mathbf{v}_g - \mathbf{v}_k) + \nabla \cdot [\boldsymbol{\tau}_k]$$

3.2.3 Constitutive Equations

$$\varepsilon_g + \varepsilon_l = 1$$

Equation of State:

$$\rho_g = \frac{\bar{M}_g P_g}{z R T_g}$$

Drag Coefficients: (Based on Ergun Equation)

$$\beta_{gk} = \beta_{kg} = 150 \frac{(1 - \varepsilon_g) \varepsilon_k \mu_g}{(\varepsilon_g d_k \psi_k)^2} + 1.75 \frac{\rho_g \varepsilon_k |v_g - v_k|}{\varepsilon_g d_k \psi_k} \quad \varepsilon_g > 0.8$$

$$\beta_{gk} = \beta_{kg} = \frac{3}{4} C_D \frac{\rho_g \varepsilon_k |v_g - v_k|}{d_k \psi_k} \varepsilon_g^{-2.65} \quad \varepsilon_g < 0.8$$

where

$$C_D = \frac{24}{Re_k} (1 + 0.15 Re_k^{0.687})$$

$$Re_k = \frac{\rho_g \varepsilon_g |v_g - v_k| d_k \psi_k}{\mu_g} \quad Re_k = 1000 \quad \text{if} \quad Re_k > 1000$$

External Forces Acting on Each Phase.

$$F_g = \frac{g}{\varepsilon_g}$$

$$F_k = \frac{g}{\varepsilon_g} \left(1 - \frac{1}{\rho_k} \sum_{m=g,k} \varepsilon_m \rho_m \right)$$

Shear Stresses.

$$[\tau_g] = \left\{ -P_g - \frac{2}{3} \mu_g \varepsilon_g \nabla \cdot v_g \right\} [I] + \mu_g \varepsilon_g [\nabla v_g + (\nabla v_g)^T]$$

$$[\tau_k] = \left\{ -P_k + \tau_{ck} + \left(\xi_k - \frac{2}{3} \mu_k \right) \nabla \cdot v_k \right\} [I] + \mu_k [\nabla v_k + (\nabla v_k)^T]$$

Cohesive Force.

$$\tau_{ck} = 10^{10.6\epsilon_k + 5.5}$$

Empirical stress modules.

$$\mu_k = 5\epsilon_k \quad \xi_k = 0$$

$$VP_k = G(\epsilon_k) \nabla \epsilon_k \quad G(\epsilon_k) = 10^{8.76\epsilon_k - 0.27}$$

3.3 Operating Conditions

The operating conditions, which cover a wide range of temperature, pressure, gas and liquid velocities, are chosen from the gas-liquid two-phase RUN E-3 series at LaPorte pilot plant (Air Products 1987).

Diameter of the reactor	59 cm
Height of the reactor	600 cm
Superficial gas velocity	0--20 cm/s
Superficial liquid velocity	0--12 cm/s
Temperature	76--250 °C
Pressure	170--6,310 kPa
Gas	N ₂
Liquid	Freeze-100
Density of liquid	0.861-0.000643xT (°C)
Number of cells in the grid	34x33
Time interval	10 ⁻⁴ sec.

3.4 Numerical Considerations

3.4.1 Initial Conditions. The initial gas hold-up and velocities are assumed to be uniform over the whole bed. A constant pressure, listed in Table 3.1, is kept at the top of the bed and isothermal is assumed in these simulation runs. The initial conditions are summerized as following:

gas	liquid
$\epsilon_g = 0.1$	$\epsilon_l = 0.9$
$u_g = 0$	$u_l = 0$
$v_g = 1 \text{ cm/s}$	$v_l = 0$
$T_g = T_0$	$T_l = T_0$
$P_g = P_0$	$P_l = P_0$

3.4.2 Boundary Conditions. The inflow conditions, including gas and liquid velocities and volume fractions, and the top pressure are prescribed. Assumption of non-slip at the walls is made for both gas and liquid phases. The boundary conditions around the computing mesh, shown as Figure 3.1, are summerized as following.

Table 3.1 Computed and Measured Gas Holdups at Reactor Height 137cm

Test #	P (kPa)	T °C	V_g cm/s	V_l m/s	ϵ_g (test)	ϵ_g (computed)	Error %
E-3-4	170	76	4	2	0.1300	0.1430	10.0
E-3-5	170	76	7	2	0.2000	0.1950	-2.5
E-3-6	170	76	13	2	0.2800	0.3030	8.2
E-3-7	170	76	20	2	0.3200	0.2789	-12.8
E-3-8	660	76	3	2	0.1800	0.1694	-5.9
E-3-9	660	76	7	2	0.2600	0.2483	-4.5
E-3-13	6310	76	3	2	0.2400	0.2403	0.1
E-3-14	6310	76	6	2	0.3800	0.3607	-5.1
E-3-15	6310	76	8	2	0.4200	0.3983	-5.2
E-3-16	6310	149	12	2	0.4900	0.4633	-5.4
E-3-17	6310	149	10	2	0.4400	0.4197	-4.6
E-3-18	6250	153	4	2	0.2500	0.2361	-5.6
E-3-20	6340	224	3	2	0.1900	0.2133	12.3
E-3-21	6310	226	6	2	0.3400	0.3820	12.4
E-3-22	6310	223	8	2	0.4500	0.5220	16.0
E-3-23	6310	221	11	2	0.4900	0.6165	25.8
E-3-28	660	77	12	2	0.3200	0.3070	-4.1
E-3-29	660	77	19	2	0.3800	0.4170	9.7
E-3-35	6310	149	10	2	0.4600	0.4825	18.1
E-3-36	6310	149	9	5	0.4300	0.4122	-4.1

Figure 3.1 Reactor Operation and Simulation Grid

		gas	liquid
bottom	gas orifice	$\varepsilon_g = 1$	$\varepsilon_l = 0$
		$v_g = v_{g0}$	$v_l = 0$
		$u_g = 0$	$u_l = 0$
	liquid orifice	$\varepsilon_g = 0$	$\varepsilon_l = 1$
		$v_g = 0$	$v_l = v_{l0}$
		$u_g = 0$	$u_l = 0$
	wall	$v_g = 0$	$v_l = 0$
		$u_g = 0$	$u_l = 0$
left/right		$v_g = 0$	$v_l = 0$
		$u_g = 0$	$u_l = 0$
top		$\frac{\partial \varepsilon_g}{\partial y} = 0$	$\frac{\partial \varepsilon_l}{\partial y} = 0$
		$\frac{\partial v_g}{\partial y} = 0$	$\frac{\partial v_l}{\partial y} = 0$
		$u_g = 0$	$u_l = 0$
		$P_g = P_0$	$P_l = P_0$

3.5 Results and Discussions

The computing time for each case listed on Table 3.1 is 20 seconds and the gas holdup is time and cross section area average value.

The axial gas holdup profiles at various superficial gas velocities are shown in Figure 3.2. Both measured at Air Products' pilot plant and computed gas hold-up profiles exhibit the trend toward higher gas volume fraction at higher elevations, specially at high superficial gas velocity. There are about 20% differences between the simulations and measurements around the lower elevations at high superficial gas velocities, because of the

Figure 3.2 Computed and Experimental Gas Holdup Profiles

significant wall effect at high gas velocity. Figure 3.2 shows that gas hold-up increases with the increasing in gas velocity also.

Figure 3.3 shows the effects of liquid velocity on the gas hold-up. Two liquid velocities, 2 cm/s and 5 cm/s were used in the simulations. The gas hold-up at higher liquid velocity is slightly less than that of the lower liquid velocity because of the introduction of more liquid. The slopes of curves in Figure 3.3 indicate a longer entrance length occurring at high liquid velocity.

As shown in Figure 3.4, the superficial gas velocity has a significant and positive effect on the gas hold-up.

Temperature has little effect on the gas hold-up since the temperature does not change the gas density as much as the pressure does (Figure 3.5).

Figure 3.6 shows the effect of pressure on gas hold-up. High gas hold-up has been observed with high pressure because the gas density is sensitive to the pressure. It is seem that high pressure compresses from leaving the system.

The measured and computed gas hold-ups at the reactor's elevation 137 cm from its bottom are listed in Table 3.1. Compared with measurements, the computed results have an error lying in 0--25% (see Figure 3.7).

In summary, the gas hold-ups, computed with a variety of operating conditions such as temperature, pressure, gas and liquid velocities, agree well with the measurements obtained at Air Products' pilot plant. The hydrodynamic model has more flexible features than those empirical correlations in predicting the gas hold-up of gas-liquid two-phase flow systems.

Figure 3.3 Computed and Experimental Axial Gas Holdup Profiles(Effect of Liquid Velocity)

Figure 3.4 Computed and Experimental Gas Holdup vs Gas Velocity (at Reactor Height=137cm)

Figure 3.5 Effect of Temperature on Gas Holdup
(at Reactor Height=137cm)

Figure 3.6 Effect of Pressure on Gas Holdup
(at Reactor Height=137cm)

Figure 3.7 Comparison of Experimental and Computed Gas Holdup

CHAPTER IV

METHANOL SYNTHESIS WITH SLURRY CIRCULATION

4.1 Introduction

The operations of liquid phase methanol process at Air Products' (1987) LaPorte pilot plant have significantly advanced development of the technology for producing methanol from coal conversion gases. Although a great deal of progress was made in demonstrating the liquid-entrained mode of operation with circulation of slurry, the RUN E-2 series also revealed that more research must be conducted if the slurry bubble column reactor is to achieve the performance of the bench-scale reactors at high slurry loadings. The hydrodynamics of the reactor deserves further study to understand the cause of the axial solid density profile observed in the bottom section of the reactor while operating with high slurry concentrations. Slurry properties such as viscosity, which may affect the reactor performance and hydrodynamics, should be determined at operating conditions.

Viking System Internationals (1993) had developed an one-dimensional, steady state model to design slurry bubble column reactors. Their model can not predict the reactor hydrodynamics. In this chapter, IIT's hydrodynamic model has been modified with methanol reaction and used to simulate slurry bubble column reactor's dynamic behaviors (WU and Gidaspow 1994).

4.2 Reaction, Kinetics and Mass transfer

4.2.1 Reaction and Kinetics. In the synthesis of methanol from syn-gas, a coal conversion gas, the following chemical reactions primarily discussed:

Several researchers had investigated the kinetic rate for methanol synthesis. There two kinds of models proposed, power law rate expressions and mechanistic models.

A mechanistic model was developed by Lee (1990):

$$r_{\text{MeOH}} = A \exp\left(-\frac{E}{RT}\right) (C_{\text{H}_2} - C_{\text{H}_2, \text{eq}})$$

$$A = 3380 \text{ cm}^3 / \text{kg} \cdot \text{s} \quad E = 18,800 \text{ kcal} / \text{kmol}$$

This model does not account for partial pressure of carbon monoxide,

Wedel et al. (1982) developed a power law rate expression based on a review of various literature models:

$$r = \epsilon_s \rho_s \left\{ 1.98 \times 10^7 \exp(-56343/RT) P_{\text{H}_2}^{0.4} P_{\text{CO}}^{0.18} - 2.15 \times 10^{10} \exp(-85930/RT) P_{\text{CH}_3\text{OH}}^{0.13} \right\}$$

This model is used in this simulation and a mechanistic model, provided by Air Products (1992) kinetics research, has been used in the simulation of chapter 5.

4.2.2 Mass Transfer Between Gas and Liquid Phase. The mass transfer rate can be expressed as:

$$R_j = \epsilon_t K_{jt} a (C_j^{g-l} - C_j^l)$$

Where C_j^l is bulk concentration of species j and C_j^{g-l} is concentration of species j at gas-liquid interface which can be defined by Henry law:

$$C_j^{g-l} = \frac{f_j}{H_j}$$

Graff et al. (1988) measured solubilities of syn-gas in the temperature of 210 to 260°C in a high molecular weight solvent. Their temperature depended Henry constants could be used with LaPorte's operating conditions.

$$H_{CO} = 0.175 \exp(638/RT)$$

$$H_{CO_2} = 0.402 \exp(-6947/RT)$$

$$H_{H_2} = 0.0782 \exp(4875/RT)$$

$$H_{CH_3OH} = 1.49 \exp(-17235/RT)$$

4.3 Species Balance

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g y_g^j) + \nabla \cdot (\epsilon_g \rho_g y_g^j \mathbf{v}_g) = \frac{\alpha^j \epsilon_s \rho_s M^j \mathbf{r}}{3.6 \times 10^6} - \epsilon_\ell M^j \mathbf{R}_j$$

Liquid Phases:

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k y_k^j) + \nabla \cdot (\epsilon_k \rho_k y_k^j \mathbf{v}_k) = \epsilon_\ell M^j \mathbf{R}_j$$

$$\sum_j^N y_k^j = 1 \quad k = g, \ell$$

j=	CO	CO ₂	H ₂	CH ₄	CH ₃ OH	wax
$\alpha^j =$	-1	0	-2	0	1	0

The phase changes, consisting of mass transfer of species between gas and liquid, will be used in the hydrodynamic models described in section 4.4:

$$\dot{m}_\ell = \epsilon_\ell \sum_j^N M^j \mathbf{R}_j \quad \dot{m}_g = -\dot{m}_\ell \quad \dot{m}_s = 0$$

4.4 Mathematical Models (Empirical)

No energy equation is needed in this simulation because of isothermal assumption. The empirical models for solid pressure and viscosity are used. The kinetic theory approach will be used in other computations depicted in chapter 5 and 6.

4.4.1 Continuity Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = \dot{m}_g$$

Liquid and Solid Phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k) = \dot{m}_k$$

4.4.2 Momentum Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \mathbf{v}_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g \mathbf{v}_g) = \epsilon_g \rho_g \mathbf{F}_g + \sum_{m=\ell, s} \beta_{gm} (\mathbf{v}_m - \mathbf{v}_g) + \nabla \cdot [\boldsymbol{\tau}_g] + \dot{m}_g \mathbf{v}_g$$

Liquid and Solid Phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k \mathbf{v}_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k \mathbf{v}_k) = \epsilon_k \rho_k \mathbf{F}_k + \sum_{m=g, \ell, s} \beta_{km} (\mathbf{v}_m - \mathbf{v}_k) + \nabla \cdot [\boldsymbol{\tau}_k] + \dot{m}_k \mathbf{v}_k$$

4.4.3 Constitutive Equations

$$\epsilon_g + \epsilon_\ell + \epsilon_s = 1$$

Equation of State:

$$\rho_g = \frac{\bar{M}_g P_g}{z R T_g}$$

Drag Coefficients: ($k=\ell, s$) (Based on Ergun Equation)

$$\beta_{gk} = \beta_{kg} = 150 \frac{(1-\epsilon_g) \epsilon_k \mu_g}{(\epsilon_g d_k \psi_k)^2} + 1.75 \frac{\rho_g \epsilon_k |\mathbf{v}_g - \mathbf{v}_k|}{\epsilon_g d_k \psi_k} \quad \epsilon_g > 0.8$$

$$\beta_{gk} = \beta_{kg} = \frac{3}{4} C_D \frac{\rho_g \epsilon_k |\mathbf{v}_g - \mathbf{v}_k|}{d_k \psi_k} \epsilon_g^{-2.65} \quad \epsilon_g < 0.8$$

$$\beta_{\ell s} = \beta_{s\ell} = \frac{3}{2}(1+e) \frac{\rho_s \rho_\ell \varepsilon_s \varepsilon_\ell |\mathbf{v}_\ell - \mathbf{v}_s|}{\rho_s d_s^3 + \rho_\ell d_\ell^3} (d_s + d_\ell)^2$$

where

$$C_D = \frac{24}{Re_k} (1 + 0.15 Re_k^{0.687})$$

$$Re_k = \frac{\rho_g \varepsilon_g |\mathbf{v}_g - \mathbf{v}_k| d_k \psi_k}{\mu_g} \quad Re_k = 1000 \quad \text{if} \quad Re_k > 1000$$

External Forces Acting on Each Phase:

$$\mathbf{F}_g = \frac{\mathbf{g}}{\varepsilon_g}$$

$$\mathbf{F}_k = \frac{\mathbf{g}}{\varepsilon_g} \left(1 - \frac{1}{\rho_k} \sum_{m=g,\ell,s} \varepsilon_m \rho_m \right) \quad k=\ell,s$$

Shear Stresses:

$$[\tau_g] = \left\{ -\mathbf{P}_g - \frac{2}{3} \mu_g \varepsilon_g \nabla \cdot \mathbf{v}_g \right\} [\mathbf{I}] + \mu_g \varepsilon_g [\nabla \mathbf{v}_g + (\nabla \mathbf{v}_g)^T]$$

$$[\tau_k] = \left\{ -\mathbf{P}_k + \tau_{ck} + \left(\xi_k - \frac{2}{3} \mu_k \right) \nabla \cdot \mathbf{v}_k \right\} [\mathbf{I}] + \mu_k [\nabla \mathbf{v}_k + (\nabla \mathbf{v}_k)^T] \quad k=\ell,s$$

$$\text{Cohesive Force:} \quad \tau_{ck} = 10^{10.6\varepsilon_k + 5.5} \quad k=\ell,s$$

Empirical Solids Pressure and Viscosity: ($k=\ell,s$)

$$\mu_k = 5\varepsilon_k \quad \xi_k = 0$$

$$\mathbf{VP}_k = \mathbf{G}(\varepsilon_k) \nabla \varepsilon_k \quad \mathbf{G}(\varepsilon_k) = 10^{8.76\varepsilon_k - 0.27}$$

4.5 Operating Conditions

The operating conditions are same as those of LaPorte's RUN E-2-B (Air Products 1987).

The operating conditions are summarized as following.

Diameter of the reactor	59 cm
Height of the reactor	600 cm
Superficial gas velocity	11.9 cm/s
Superficial liquid velocity	4.9 cm/s
Superficial solid velocity	4.9 cm/s
Temperature	250 °C
Pressure	6,310 kPa
Catalyst diameter	50 μm
Catalyst concentration	46.06%(wt)
Liquid	wax
Density of liquid	0.70025 g/cm ³
Number of cells in the grid	18x33
Time interval	10 ⁻⁴ sec.

4.6 Numerical Considerations

4.6.1 Initial Conditions. The initial gas, liquid and solid volume fractions and velocities are assumed to be uniform over the whole bed. A constant pressure, 6.31 MPa, is kept at the top of the bed and isothermal is assumed in these simulation runs. The initial conditions are summerized as followings:

gas	liquid	solid
$\varepsilon_g=0.3800$	$\varepsilon_l=0.4805$	$\varepsilon_s=0.1395$
$u_g=0$	$u_l=0$	$u_s=0$
$v_g=31.3\text{cm/s}$	$v_l=7.9\text{cm/s}$	$v_s=7.9\text{cm/s}$

Syn-gas composition:

	CO	CO ₂	H ₂	CH ₄	CH ₃ OH
%mol	19.33	4.05	59.34	17.05	0.24
y_g^j %wt	48.39	15.93	10.61	24.38	0.69

4.6.2 Boundary Conditions. The inflow conditions, including gas, liquid and solid velocities and volume fractions, and the top pressure are prescribed. Assumption of non-slip at the walls is made for both gas and liquid phases. The boundary conditions around the computing mesh, shown as Figure 4.1, are summerized as following.

	gas	liquid	solid
bottom	$\varepsilon_g=0.3800$	$\varepsilon_l=0.4805$	$\varepsilon_s=0.1395$
	$v_g=31.3\text{cm/s}$	$v_l=7.9\text{cm/s}$	$v_s=7.9\text{cm/s}$
	$u_g=0$	$u_l=0$	$u_s=0$
	$y_g^j=y_g^{j0}$	$y_l^j=0$	$y_s^j=0$
left/right	$v_g=0$	$v_l=0$	$v_s=0$
	$u_g=0$	$u_l=0$	$u_s=0$
top	$\frac{\partial \varepsilon_g}{\partial y}=0$	$\frac{\partial \varepsilon_l}{\partial y}=0$	$\frac{\partial \varepsilon_s}{\partial y}=0$
	$\frac{\partial v_g}{\partial y}=0$	$\frac{\partial v_l}{\partial y}=0$	$\frac{\partial v_s}{\partial y}=0$
	$u_g=0$	$u_l=0$	$u_s=0$
	$\frac{\partial y_g^j}{\partial y}=0$	$\frac{\partial y_l^j}{\partial y}=0$	$\frac{\partial y_s^j}{\partial y}=0$

Figure 4.1 Reactor Operation with Slurry Circulation and Simulation Grid

4.6.3 Computation. Finite difference method is used to solve these partial differential equations simultaneously and the values of following variables at each time step and cell position can be obtained.

$$\varepsilon_g, \varepsilon_l, \varepsilon_s, v_g, v_l, v_s, P_g, y_g^j, y_l^j, y_s^j$$

$$j = \text{CO}, \text{CO}_2, \text{H}_2, \text{CH}_4, \text{CH}_3\text{OH}, \text{wax}$$

4.7 Computational Results and Discussions

The computing run was performed on a HP735 Workstation and the simulation time is 40 seconds. All data are taken by time and cross section area average over the period of last ten seconds.

Table 4.1 lists the material balance of the simulations and Table 4.2 lists the comparison of simulation and LaPorte's RUN E-2-B. They show very close CO conversions. The computed rate of methanol production is a little higher than that of RUN E-2-B since the kinetic model does not reflect exactly reactor performance.

Table 4.1 Material Balance: (CO:H₂=1:3)

	Inlet		Outlet	
	% mol	kgmol/hr	% mol	kgmol/hr
CO	19.33	33.7	16.40	25.9
CO ₂	4.05	7.0	4.40	7.0
H ₂	59.34	103.3	55.50	87.5
CH ₃ OH	0.24	0.4	5.00	7.9
CH ₄	17.05	29.5	18.70	29.5
$\bar{M}$ (kg/kgmol)	11.18		12.23	
Flowrate (kgmol/hr)		173.9		157.8
Flowrate (kg/hr)		1945		1930

Table 4.2 Comparison of Simulation and Air Products' (1987) RUN E-2-B:

	CO conv. (%)	Slurry conc. (%wt)	Total catalyst (kgmole/hr)	CH ₃ OH (gmol/hr/kg)	Net CH ₃ OH (TPD)
Simulation	22.3	46.4	7.9	10.88	6.07
RUN E-2-B	21.3	48.9	7.3	9.9	5.60

Figure 4.2 shows a comparison of computed and measured catalyst concentrations for LaPorte reactor. At zero time the concentration was uniform. After 20 seconds the density at the bottom of the reactor increased substantially. The computed and experimental profiles match reasonably well.

Figure 4.3 shows the computed axial profiles for gas, liquid and solid three phases; Figure 4.4 shows the methanol concentration profiles in both gas and liquid phases.

Figure 4.5a--c show a series of catalyst concentration distribution as a function of time. After startup, catalyst-free voidage are forming and propagating with time elasting. These voidages significantly reduce the reactor performance.

Figure 4.6a--c show a series of methanol concentration propagation as a function of time. The concentration is getting dense until a final steady state.

In summary, the computed catalyst concentration profile and the rate of methanol production agree well with the measurements obtained at LaPorte pilot plant. The dynamic simulation results such as the time series catalyst distributions are very helpful for the understanding of the reactor hydrodynamic behavior.

Figure 4.2 Computed and Experimental Axial Catalyst Volume Fraction Profiles

Figure 4.3 Computed Gas, Liquid and Solid
Axial Volume Fraction Profiles

Figure 4.4 Computed Axial Methanol Concentration Profile

Figure 4.5a The Catalyst Concentration Profiles as a Function of Time

Figure 4.5b The Catalyst Concentration Profiles as a Function of Time

Figure 4.5c The Catalyst Concentration Profiles as a Function of Time

Figure 4.6a The Methanol Concentration Profiles as a Function of Time

Figure 4.6b The Methanol Concentration Profiles as a Function of Time

Figure 4.6c The Methanol Concentration Profiles as a Function of Time

CHAPTER V

METHANOL SYNTHESIS WITHOUT SLURRY CIRCULATION

5.1 Introduction

Liquid-entrained operations at the LaPorte liquid phase methanol process development unit continued with a simplified and optimized process (Air Products 1992). The process with circulation of slurry described in previous chapter was no longer in use since costly operation. A series of operations RUN E-8 was conducted in a process without circulation of slurry. This chapter described the simulations for the case RUN E-8.1 (Gidaspow et al. 1995). An internal heat exchanger was considered in this computation and kinetic theory approach is used to compute the solid viscosity.

In order to examine the convergence of the computational techniques, an additional simulation run was made with same operating conditions except of the grid size.

5.2 Reaction and Kinetics Model

To predict a more accurate methanol production rate, a power law rate expression, depicted in Air Products' (1992) report, is used in this computational run. The model was based on the rate expressions for methanol synthesis (Weimer et al. 1987) and the parameters were determined from experimental research.

$$r = \varepsilon_p K_r f_{CO}^{1/3} f_{H_2}^{2/3} \left(1 - \frac{f_{MEOH}}{K_{eq} f_{CO} f_{H_2}^2} \right)$$

The mass transfer and species balances are same as that described in chapter 4.

5.3 Mathematical Models (Kinetic Theory)

Energy equation is needed in this simulation because of non-isothermal situation with internal heat exchanger. The kinetic theory approach is used for computing the solid pressure and viscosity.

5.3.1 Continuity Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = \dot{m}_g$$

Liquid and Solid Phase: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k) = \dot{m}_k$$

5.3.2 Momentum Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \mathbf{v}_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g \mathbf{v}_g) = \epsilon_g \rho_g \mathbf{F}_g + \sum_{m=\ell, s} \beta_{gm} (\mathbf{v}_m - \mathbf{v}_g) + \nabla \cdot [\boldsymbol{\tau}_g] + \dot{m}_g \mathbf{v}_g$$

Liquid and Solid Phase: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k \mathbf{v}_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k \mathbf{v}_k) = \epsilon_k \rho_k \mathbf{F}_k + \sum_{m=g, \ell, s} \beta_{km} (\mathbf{v}_m - \mathbf{v}_k) + \nabla \cdot [\boldsymbol{\tau}_k] + \dot{m}_k \mathbf{v}_k$$

5.3.3 Energy Equations

Gas Phase:

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_g \rho_g H_g) + \nabla \cdot (\epsilon_g \rho_g H_g \mathbf{v}_g) = & \left(\frac{\partial P_g}{\partial t} + P_g \nabla \cdot \mathbf{v}_g \right) + \nabla \cdot (\mathbf{k}_g \nabla T_g) + \Phi_g + \sum_i \mathbf{r}_{ig} \Delta H_{ig} \\ & + \sum_{m=\ell, s} \left\{ h_{vm} (T_m - T_g) + \beta_{gm} (\mathbf{v}_m - \mathbf{v}_g)^2 - \dot{m}_m H_g \right\} \end{aligned}$$

Liquid and Solid Phase: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\varepsilon_k \rho_k H_k) + \nabla \cdot (\varepsilon_k \rho_k H_k \mathbf{v}_k) = h_{vk}(T_g - T_k) + \sum_{m=g,\ell,s} \beta_{km} (v_m - v_k)^2 + \dot{m}_k H_g \\ + \nabla \cdot (k_k \nabla T_k) + \sum_i r_{ik} \Delta H_{ik} + \Phi_k$$

5.3.4 Fluctuating Energy Equation: (k=ℓ,s)

$$\frac{3}{2} \left[\frac{\partial}{\partial t}(\varepsilon_k \rho_k \Theta_k) + \nabla \cdot (\varepsilon_k \rho_k \Theta_k \mathbf{v}_k) \right] = \nabla \cdot (\kappa_k \nabla \Theta_k) - \gamma_k - 3\beta_{kg} \Theta_k + \Phi_k$$

5.3.5 Constitutive Equations

$$\varepsilon_g + \varepsilon_\ell + \varepsilon_s = 1$$

Equation of State:

$$\rho_g = \frac{\overline{M}_g P_g}{z R T_g}$$

Drag Coefficients: (k=ℓ,s) (Based on Ergun Equation)

$$\beta_{gk} = \beta_{kg} = 150 \frac{(1 - \varepsilon_g) \varepsilon_k \mu_g}{(\varepsilon_g d_k \psi_k)^2} + 1.75 \frac{\rho_g \varepsilon_k |v_g - v_k|}{\varepsilon_g d_k \psi_k} \quad \varepsilon_g > 0.8$$

$$\beta_{gk} = \beta_{kg} = \frac{3}{4} C_D \frac{\rho_g \varepsilon_k |v_g - v_k|}{d_k \psi_k} \varepsilon_g^{-2.65} \quad \varepsilon_g < 0.8$$

$$\beta_{\ell s} = \beta_{s\ell} = \frac{3}{2} (1 + e) \frac{\rho_s \rho_\ell \varepsilon_s \varepsilon_\ell |v_\ell - v_s|}{\rho_s d_s^3 + \rho_\ell d_\ell^3} (d_s + d_\ell)^2$$

where

$$C_D = \frac{24}{Re_k} (1 + 0.15 Re_k^{0.687})$$

$$Re_k = \frac{\rho_g \varepsilon_g |v_g - v_k| d_k \psi_k}{\mu_g}$$

$$Re_k = 1000 \quad \text{if} \quad Re_k > 1000$$

External Forces Acting on Each Phase

$$F_g = \frac{g}{\varepsilon_g}$$

$$F_k = \frac{g}{\varepsilon_g} \left(1 - \frac{1}{\rho_k} \sum_{m=g, \ell, s} \varepsilon_m \rho_m \right) \quad k = \ell, s$$

Enthalpy:

$$H_g = C_{p_g} (T_g - T_g^0)$$

$$H_k = C_{p_k} (T_k - T_k^0)$$

Gas-Solid Heat Transfer: (Gunn's Model)

$$Nu_k = \left\{ (2 + 5\varepsilon_k^2)(1 + 0.7Re_k^{0.2}Pr^{1/3}) + \left(\frac{2}{15} + 1.2\varepsilon_k^2 \right) Re_k^{0.7}Pr^{1/3} \right\} Sp_k$$

$$Nu_k = \frac{h_{vk} d_k}{k_g^0} \quad Pr = \frac{C_{p_g} \mu_g}{k_g^0} \quad Sp_k = \frac{6\varepsilon_k}{d_k}$$

Gas Phase Heat Transfer:

$$k_g^0 = 8.65 \times 10^5 \left(\frac{T_g}{1400} \right)^{1.786}$$

$$k_g = (1 - \sqrt{1 - \varepsilon_g}) k_g^0$$

Particulate Phase Heat Transfer:

$$\frac{k_k}{k_g^0} = \frac{\sqrt{\varepsilon_s}}{(1 - \varepsilon_g)} \left\{ \phi \frac{k_k^*}{k_g^0} + (1 - \phi) \frac{k_k^0}{k_g^0} \right\}$$

$$\frac{k_k^0}{k_g^0} = \frac{2}{A_k} \left\{ \frac{B_k \{ (k_k^*/k_g^0) - 1 \}}{A_k^2 (k_k^*/k_g^0)} \ln \frac{(k_k^*/k_g^0)}{B_k} \frac{B_k - 1}{A_k} \frac{B_k + 1}{2} \right\}$$

where

$$A_k = 1 - \frac{B_k}{(k_k^*/k_g^0)}$$

$$B_k = 1.25 \left(\frac{\varepsilon_k}{\varepsilon_g} \right)^{10/9} (1 + 3\chi)$$

$$\chi = \sqrt{\frac{(\sum \varepsilon_k \rho_k)(\sum \varepsilon_k \rho_k / d_k^2)}{(\sum \varepsilon_k \rho_k / d_k)^2}} - 1 \quad k=l,s$$

$$k_k^* = 0.3289 \quad \varphi = 7.26 \times 10^{-3}$$

Shear Stresses:

$$[\tau_g] = \left\{ -P_g - \frac{2}{3} \mu_g \varepsilon_g \nabla \cdot \mathbf{v}_g \right\} [\mathbf{I}] + \mu_g \varepsilon_g [\nabla \mathbf{v}_g + (\nabla \mathbf{v}_g)^T]$$

$$[\tau_k] = \left\{ -P_k + \tau_{ck} + \left(\xi_k - \frac{2}{3} \mu_k \right) \nabla \cdot \mathbf{v}_k \right\} [\mathbf{I}] + \mu_k [\nabla \mathbf{v}_k + (\nabla \mathbf{v}_k)^T] \quad k=l,s$$

Cohesive Force:

$$\tau_{ck} = 10^{10.6\varepsilon_k + 5.5} \quad k=l,s$$

Solids Pressure and Viscosity Defined by Granular Temperature: ($k=l,s$)

$$P_k = \rho_k \varepsilon_k \Theta_k \{ 1 + 2(1 + e_k) g_{0k} \varepsilon_k \}$$

$$\xi_k = \frac{4}{3} \varepsilon_k^2 \rho_k d_k g_{0k} (1 + e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

$$\mu_k = \frac{2\mu_{k_{at}}}{(1 + e_k) g_{0k}} \left\{ 1 + \frac{4}{5} (1 + e_k) g_{0k} \varepsilon_k \right\}^2 + \frac{4}{5} \varepsilon_k^2 \rho_k d_k g_{0k} (1 + e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

$$\gamma_k = 3(1 - e_k^2) \varepsilon_k^2 \rho_k g_{0k} \Theta_k \left\{ \frac{4\sqrt{\Theta_k}}{d_k \sqrt{\pi}} - \nabla \cdot \mathbf{v}_k \right\}$$

$$\kappa_k = \frac{2\kappa_{k_{at}}}{(1 + e_k) g_{0k}} \left\{ 1 + \frac{6}{5} (1 + e_k) g_{0k} \varepsilon_k \right\}^2 + 2\varepsilon_k^2 \rho_k d_k g_{0k} (1 + e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

Where

$$g_{0k} = \left\{ 1 - \left(\frac{\varepsilon_k}{\varepsilon_{k_{\max}}} \right)^{1/3} \right\}^{-1} \quad \mu_{k_{\text{eff}}} = \frac{5}{96} \rho_k d_k \sqrt{\pi \Theta_k} \quad \kappa_{k_{\text{eff}}} = \frac{75}{384} \rho_k d_k \sqrt{\pi \Theta_k}$$

Energy Production Rate:

$$\Phi_k = [\tau_k] : \nabla \mathbf{v}_k \quad \mathbf{k} = \mathbf{g}, \ell, \mathbf{s}$$

5.4 Operating Conditions

The operating conditions are same as those of LaPorte's RUN E-8.1 (Air Products 1991). Figure 5.1 shows reactor grid employed in this simulation. Gas passes through and slurry stays in the reactor.

Diameter of the reactor	57cm
Height of the reactor	813cm
Superficial gas velocity	15.24 cm/s
Superficial liquid velocity	0 cm/s
Superficial solid velocity	0 cm/s
Temperature	250.3 °C
Pressure	753.2 psig
Catalyst diameter	50 μm
Liquid	wax
Density of liquid	0.70025 g/cm ³
Grid size (dx & dy)	1.68cm x 20.3 cm
Number of cells in the grid	34x40
Time interval	10 ⁻⁴ sec.

Figure 5.1 Reactor Operation without Slurry Circulation and Simulation Grid

5.5 Numerical Considerations

5.5.1 Initial Conditions. A constant pressure, 753.2 psig, is kept at the top of the reactor. The initial conditions are summerized as followings:

	gas	liquid	solid
0 – 183	$\epsilon_g = 0.050$	$\epsilon_l = 0.426$	$\epsilon_s = 0.524$
	$v_g = 4.8$	$v_l = 0$	$v_s = 0$
	$u_g = 0$	$u_l = 0$	$u_s = 0$
183 – 427	$\epsilon_g = 0.050$	$\epsilon_l = 0.950$	$\epsilon_s = 0.000$
	$v_g = 4.8$	$v_l = 0$	$v_s = 0$
	$u_g = 0$	$u_l = 0$	$u_s = 0$
427 – 813	$\epsilon_g = 1.000$	$\epsilon_l = 0.000$	$\epsilon_s = 0.000$
	$v_g = 0.24$	$v_l = 0$	$v_s = 0$
	$u_g = 0$	$u_l = 0$	$u_s = 0$

Syn-gas composition (CO-rich):

	CO	CO ₂	H ₂	N ₂	CH ₃ OH
%mol	51	13	35	1	0
y_g^j %wt	68.07	27.26	3.34	1.33	0

5.5.2 Boundary Conditions. The thermal boundary conditions around the computing mesh and a heat block are summerized as following.

Boundaries:

$$\frac{\partial T_k}{\partial x} = 0$$

$$\frac{\partial T_k}{\partial y} = 0$$

$$k = g, \ell, k$$

Around heat block:

**Case 1. constant
heat flux**

$$\frac{\partial T_k}{\partial x} = C_{kx} \cdot \varepsilon_k$$

$$\frac{\partial T_k}{\partial y} = C_{ky} \cdot \varepsilon_k$$

$$k = g, \ell, k$$

**Case 2. constant block
temperature T_b**

$$\frac{\partial T_k}{\partial x} = K_x \cdot \varepsilon_k (T_b - T_k)$$

$$\frac{\partial T_k}{\partial y} = K_y \cdot \varepsilon_k (T_b - T_k) \quad k = g, \ell, k$$

The inflow conditions, including gas velocity and composition, and the top pressure are prescribed. Assumption of non-slip at the walls is made for both gas and liquid phases. The boundary conditions around the computing mesh, shown as Figure 5.1, are summerized as followings.

	gas	liquid	solid
bottom	$\varepsilon_g = 1$	$\varepsilon_\ell = 0$	$\varepsilon_s = 0$
orifice	$v_g = 81.3 \text{ cm/s}$	$v_\ell = 0$	$v_s = 0$
	$u_g = 0$	$u_\ell = 0$	$u_s = 0$
	$y_g^j = y_g^{j0}$	$y_\ell^j = 0$	$y_s^j = 0$
left/right	$v_g = 0$	$v_\ell = 0$	$v_s = 0$
	$u_g = 0$	$u_\ell = 0$	$u_s = 0$
top	$\frac{\partial \varepsilon_g}{\partial y} = 0$	$\frac{\partial \varepsilon_\ell}{\partial y} = 0$	$\frac{\partial \varepsilon_s}{\partial y} = 0$
	$\frac{\partial v_g}{\partial y} = 0$	$\frac{\partial v_\ell}{\partial y} = 0$	$\frac{\partial v_s}{\partial y} = 0$
	$u_g = 0$	$u_\ell = 0$	$u_s = 0$
	$\frac{\partial y_g^j}{\partial y} = 0$	$\frac{\partial y_\ell^j}{\partial y} = 0$	$\frac{\partial y_s^j}{\partial y} = 0$

BLANK PAGE

5.5.3 Computation. Finite difference method is used to solve these partial differential equations simultaneously and the values of following variables at each time step and cell position can be obtained.

$$\varepsilon_g, \varepsilon_l, \varepsilon_s, v_g, v_l, v_s, P_g, y_g^j, y_l^j, y_s^j$$

$$j = \text{CO, CO}_2, \text{H}_2, \text{N}_2, \text{CH}_3\text{OH, wax}$$

5.6 Computational Results and Discussions

The initial state of the reactor is shown in Figure 5.1. The obstacles shown represent heat exchangers. At time zero, the syn-gas is injected to reactor through six orifices and the gas velocity is increased gradually to the final value during the first computing second. The total computing time is 80 seconds. The computing run was performed on a HP735 Workstation and the simulation time is 80 seconds. All values using in the follow figures and Tables are time and cross section area average over last 40 seconds.

Table 5.1 lists the material balance of the simulations and Table 5.2 lists the comparison of simulation and LaPorte's RUN E-8.1. They show very close CO conversion and rate of methanol production.

Figure 5.2 shows the time response of gas mass flowrate at reactor top. There are strong oscillations occurring during the first 4 seconds after start-up (syn-gas injection). After that, a hydrodynamic steady state could be assumed.

Figure 5.3 shows the time average gas-liquid-solid volume fraction profiles. Table 5.2 and Figure 5.3 also show that the height of slurry and gas holdup are roughly agree with RUN E-8.1.

Table 5.1 Material Balance: (CO:H₂=1.5:1)

	Inlet		Outlet	
	% mol	kgmol/hr	% mol	kgmol/hr
CO	51	87.9	51.18	74.8
CO ₂	13	22.4	15.39	22.4
H ₂	35	60.3	23.03	33.6
CH ₃ OH	0	0.0	9.20	13.4
N ₂	1	1.7	1.18	1.7
$\bar{M}$ (kg/kgmol)	20.98		24.84	
Flowrate (kgmol/hr)		172.3		145.9
Flowrate (kg/hr)		3614		3625

Table 5.2 Comparison of Simulation and Air Products' (1991) RUN E-8.1:

	CO conv. (%)	gas holdup (%)	slurry height (inches)	total catalyst (kg)	CH ₃ OH (gmol/hr /kg)	net CH ₃ OH (TPD)
Simulation	14.24	26.9	215	740	16.93	9.62
RUN E-8.1	13.50	29.5	200	567	20.50	10.03

Figure 5.4 shows the comparison of methanol profile computed by (1993), one-dimensional steady state model and the two-dimensional hydrodynamic model. The one-dimensional model predicts a high methanol production because of the assumption of uniform catalyst concentration. The hydrodynamic model, however, predicts much close methanol production (Table 5.2) since it accounts the effect of the distribution of gas, catalyst concentrations, temperature and the flow patterns inside the reactor.

Figure 5.5a--d show the gas volume fraction distributions as a function of time. During the first 4 seconds after startup (syn-gas injection), irregular distributions reflect that the operation is non-stable which has been shown in Figure 5.2. After 4 seconds the

Figure 5.2 Gas Mass Flowrate at Reactor

Figure 5.3 Time average G-L-S Volume Fraction

Figure 5.4 Methanol Concentration Profiles from IIT's Hydrodynamic, One-dimensional and Vikings' Models

Figure 5.5a The Gas Volume Fraction Profiles as a Function of Time

Figure 5.5b The Gas Volume Fraction Profiles as a Function of Time

Figure 5.5c The Gas Volume Fraction Profiles as a Function of Time

Figure 5.5d The Gas Volume Fraction Profiles as a Function of Time

operation approaches steady state in hydrodynamic sense. The gas volume fraction decreases along the elevation because of the reaction of methanol synthesis. These figures also demonstrate a non-uniform radial distribution that one-dimensional model could not predict.

Figure 5.6a--d show the catalyst volume fraction distributions varying with time. The catalyst expands rapidly after injection of syn-gas and settles down at steady state.

Figure 5.7a-e show the methanol concentration distributions varying with time. After injection of syn-gas to the methanol free reactor from six orifices, the methanol is forming, propagating rapidly and accumulating and its concentration reaches a steady value, 9% mole, which agrees with LaPorte's RUN E-8.1.

Figure 5.8a-c show the temperature distributions varying with time. The heat exchangers absorb heat released from the exothermal methanol synthesis reaction.

Figure 5.9 shows a transient distribution of the rate of methanol reaction.

Figure 5.10 and 5.11 show solid viscosity and granular temperature distributions. The viscosity and granular temperature are high at the reactor entrance due to the injection of syn-gas.

Figure 5.12 shows a transient gas flow pattern. There exists vortices inside the reactor and they imply the good mixing feature of slurry bubble column reactor.

Figure 5.13 shows a series of gas flow patterns as a function of time.

5.7 Convergence Check

The IIT's hydrodynamic code uses the finite difference method to solve the partial differential equations. The grid size is very sensitive to the stability and convergence of

Figure 5.6a The Solid Volume Fraction Profiles as a Function of Time

Figure 5.6b The Solid Volume Fraction Profiles as a Function of Time

Figure 5.6c The Solid Volume Fraction Profiles as a Function of Time

Figure 5.6d The Solid Volume Fraction Profiles as a Function of Time

Figure 5.7a The Methanol Concentration Profiles as a Function of Time

Figure 5.7b The Methanol Concentration Profiles as a Function of Time

Figure 5.7c The Methanol Concentration Profiles as a Function of Time

Figure 5.7d The Methanol Concentration Profiles as a Function of Time

Figure 5.7e The Methanol Concentration Profiles as a Function of Time

Figure 5.8a The Temperature Profiles as a Function of Time

Figure 5.8b The Temperature Profiles as a Function of Time

Figure 5.8c The Temperature Profiles as a Function of Time

Figure 5.9 Instant Rate of Reaction (gmol/s/cm3)

Figure 5.10 Solid Viscosity Distribution

Figure 5.11 Solid Granular Temperature

Figure 5.12 Gas Transient Flow Pattern

Figure 5.13 Time Series of Gas Flow Patterns

solutions. The finer the grid is, the more accurate the solution is. Under the limits of the computer speed and the capacity of storage, one can not choose an infinite fine grid. A reasonable grid needs to be chosen carefully without losing the accuracy and convergence of the solution. The usual way to check the convergence is to solve a problem with different grid size. To do so, a simulation run, based on the previous run for Air Products' reactor (Wu et al. 1995 and Pape et al. 1996), with fine grid was conducted using IIT's hydrodynamic code. The conditions of this run are exact same as the previous run except of the grid size, in which the Δy of this run is one third less than that of the previous one.

Figure 5.14 shows the comparison of the time-averaged g-l-s volume fraction profiles in both runs. They have same slurry height, 530 cm. The curves of both runs demonstrate same profiles except of a minor difference.

Figure 5.15 and 5.16 show the frequency responses of the gas flowrate at the outlet. Both runs have approximately same frequency, 0.7 Hertz.

Figure 5.17 a&b and 5.18 a&b demonstrate the similar flow patterns in both runs that display down-flows at the walls, up-flow at the center and vortices near the bottom.

In summary, the computed slurry height, gas holdup and the rate of methanol production agree well with LaPorte's RUN E-8.1. The two-dimensional hydrodynamic model is better than the one-dimensional, steady state models in predicting the methanol production and hydrodynamics of slurry bubble column reactors.

Two grids are used in the simulations of the Air Products' reactor and a good agreement, indicating the accuracy and convergence of the simulations, is reached.

Figure 5.14 Time Average G-L-S Volume Fraction Profiles in Different Grid

Figure 5.15 Gas Flowrate at Reactor Top and Frequency Response (Fine Grid)

Figure 5.16 Gas Flowrate at Reactor Top and Frequency Response (Coarse Grid)

$t=0$ sec.

$t=1$ sec.

$t=2$ sec.

Figure 5.17a Gas Flow Patterns (Fine Grid)

Figure 5.17b Gas Flow Patterns (Fine Grid)

$t=0$ sec.

$t=1$ sec.

$t=2$ sec.

Figure 5.18a Gas Flow Patterns (Coarse Grid)

Figure 5.18b Gas Flow Patterns (Coarse Grid)

APPENDIX A
NUMERICAL BASIS OF PROGRAM MFREK

There is no way to obtain analytical solution for the hydrodynamic models described in chapter 2 unless a lot of simplizations are applied to the models and its applications are very limited. With high speed supercomputer, a very accurately numerical solution could be obtained. In this study, a program MFREK, based on the hydrodynamic models, is developed to obtain numerical solution for multiphase flow systems, using supercomputer. Appendix A describes the numerical scheme while appendix B describes the organization of the program and appendix C lists the source Fortran code and a sample input data of the program.

A.1 The Governing Equations

We will consider a multiphase system consisting of several particulate phases and a continuous phase. In the following equations, gas is treated as continuous phase, but liquid could be treated as continuous phase too in some particular situations such as liquid-solid system. The particulate phases could be one or more disperse phases, for instance, liquid and catalyst in gas-liquid-solid system of chapter 4 and 5, coal and pyrite in a gas-solid-solid system of chapter 6.

A.1.1 Continuity Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = \dot{m}_g$$

Liquid and Solid phases: ($k=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k) = \dot{m}_k$$

$$\epsilon_g + \epsilon_\ell + \epsilon_s = 1$$

A.1.2 Momentum Equations

Gas Phase:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \mathbf{v}_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g \mathbf{v}_g) = \epsilon_g \rho_g \mathbf{F}_g + \sum_{m=\ell, s} \beta_{gm} (\mathbf{v}_m - \mathbf{v}_g) + \nabla \cdot [\boldsymbol{\tau}_g] + \dot{\mathbf{m}}_g \mathbf{v}_g$$

Liquid and Solid Phases: ($\mathbf{k}=\ell, s$)

$$\frac{\partial}{\partial t}(\epsilon_k \rho_k \mathbf{v}_k) + \nabla \cdot (\epsilon_k \rho_k \mathbf{v}_k \mathbf{v}_k) = \epsilon_k \rho_k \mathbf{F}_k + \sum_{m=g, \ell, s} \beta_{km} (\mathbf{v}_m - \mathbf{v}_k) + \nabla \cdot [\boldsymbol{\tau}_k] + \dot{\mathbf{m}}_k \mathbf{v}_k$$

A.1.3 Energy Equations

Gas Phase:

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_g \rho_g H_g) + \nabla \cdot (\epsilon_g \rho_g H_g \mathbf{v}_g) &= \left(\frac{\partial P_g}{\partial t} + P_g \nabla \cdot \mathbf{v}_g \right) + \nabla \cdot (\mathbf{k}_g \nabla T_g) + \Phi_g + \sum_i \mathbf{r}_{ig} \Delta H_{ig} \\ &+ \sum_{m=\ell, s} \left\{ h_{vm} (T_m - T_g) + \beta_{gm} (\mathbf{v}_m - \mathbf{v}_g)^2 - \dot{\mathbf{m}}_m H_g \right\} \end{aligned}$$

Liquid and Solid Phases: ($\mathbf{k}=\ell, s$)

$$\begin{aligned} \frac{\partial}{\partial t}(\epsilon_k \rho_k H_k) + \nabla \cdot (\epsilon_k \rho_k H_k \mathbf{v}_k) &= h_{vk} (T_g - T_k) + \sum_{m=g, \ell, s} \beta_{km} (\mathbf{v}_m - \mathbf{v}_k)^2 + \dot{\mathbf{m}}_k H_g \\ &+ \nabla \cdot (\mathbf{k}_k \nabla T_k) + \sum_i \mathbf{r}_{ik} \Delta H_{ik} + \Phi_k \end{aligned}$$

A.2 Constitutive Equations

A.2.1 Equation of State.

$$\rho_g = \frac{\bar{M}_g P_g}{z R T_g}$$

A.2.2 Drag Coefficients

$$\beta_{gk} = \beta_{kg} = 150 \frac{(1 - \epsilon_g) \epsilon_k \mu_g}{(\epsilon_g d_k \psi_k)^2} + 1.75 \frac{\rho_g \epsilon_k |\mathbf{v}_g - \mathbf{v}_k|}{\epsilon_g d_k \psi_k} \quad \epsilon_g > 0.8$$

$$\beta_{gk} = \beta_{kg} = \frac{3}{4} C_D \frac{\rho_g \varepsilon_k |\mathbf{v}_g - \mathbf{v}_k|}{d_k \psi_k} \varepsilon_g^{-2.65} \quad \varepsilon_g < 0.8$$

where
$$C_D = \frac{24}{Re_k} (1 + 0.15 Re_k^{0.687})$$

$$Re_k = \frac{\rho_g \varepsilon_g |\mathbf{v}_g - \mathbf{v}_k| d_k \psi_k}{\mu_g} \quad Re_k = 1000 \quad \text{if} \quad Re_k > 1000$$

$$\beta_{ls} = \beta_{sl} = \frac{3}{2} (1+e) \frac{\rho_s \rho_l \varepsilon_s \varepsilon_l |\mathbf{v}_l - \mathbf{v}_s|}{\rho_s d_s^3 + \rho_l d_l^3} (d_s + d_l)^2$$

A.2.3 External Forces Acting on Each Phase

$$\mathbf{F}_g = \frac{\mathbf{g}}{\varepsilon_g}$$

$$\mathbf{F}_k = \frac{\mathbf{g}}{\varepsilon_g} \left(1 - \frac{1}{\rho_k} \sum_{m=g,l,s} \varepsilon_m \rho_m \right) + q_k \mathbf{E} \quad k=l,s$$

A.2.4 Shear Stresses

$$[\tau_g] = \left\{ -\mathbf{P}_g - \frac{2}{3} \mu_g \varepsilon_g \nabla \cdot \mathbf{v}_g \right\} [\mathbf{I}] + \mu_g \varepsilon_g [\nabla \mathbf{v}_g + (\nabla \mathbf{v}_g)^T]$$

$$[\tau_k] = \left\{ -\mathbf{P}_k + \tau_{ck} + \left(\xi_k - \frac{2}{3} \mu_k \right) \nabla \cdot \mathbf{v}_k \right\} [\mathbf{I}] + \mu_k [\nabla \mathbf{v}_k + (\nabla \mathbf{v}_k)^T] \quad k=l,s$$

$$\tau_{ck} = 10^{10.6\varepsilon_k + 5.5} \quad k=l,s$$

Empirical Solids Viscosity and Stress Model (optional, a substitution to kinetic theory).

$$\mu_k = 5\varepsilon_k \quad \xi_k = 0$$

$$\mathbf{VP}_k = \mathbf{G}(\varepsilon_k) \nabla \varepsilon_k \quad \mathbf{G}(\varepsilon_k) = 10^{8.76\varepsilon_k - 0.27}$$

A.2.5 Enthalpy

$$H_g = C_{p_g} (T_g - T_g^0)$$

$$H_k = C_{p_k} (T_k - T_k^0)$$

A.2.6 Gas-Solid Heat Transfer. (Gunn's Model)

$$Nu_k = \left\{ (2 + 5\varepsilon_k^2)(1 + 0.7Re_k^{0.2}Pr^{1/3}) + \left(\frac{2}{15} + 1.2\varepsilon_k^2\right)Re_k^{0.7}Pr^{1/3} \right\} Sp_k$$

$$Nu_k = \frac{h_{vk} d_k}{k_g^0} \quad Pr = \frac{C_{p_g} \mu_g}{k_g^0} \quad Sp_k = \frac{6\varepsilon_k}{d_k}$$

A.2.7 Gas Phase Heat Transfer

$$k_g^0 = 8.65 \times 10^5 \left(\frac{T_g}{1400} \right)^{1.786}$$

$$k_g = (1 - \sqrt{1 - \varepsilon_g}) k_g^0$$

A.2.8 Particulate Phase Heat Transfer

$$\frac{k_k}{k_g^0} = \frac{\sqrt{\varepsilon_s}}{(1 - \varepsilon_g)} \left\{ \phi \frac{k_k^*}{k_g^0} + (1 - \phi) \frac{k_k^0}{k_g^0} \right\}$$

$$\frac{k_k^0}{k_g^0} = \frac{2}{A_k} \left\{ \frac{B_k \left[(k_k^*/k_g^0) - 1 \right]}{A_k^2 (k_k^*/k_g^0)} \cdot \ln \frac{(k_k^*/k_g^0)}{B_k} \frac{B_k - 1}{A_k} \frac{B_k + 1}{2} \right\}$$

where

$$A_k = 1 - \frac{B_k}{(k_k^*/k_g^0)}$$

$$B_k = 1.25 \left(\frac{\varepsilon_k}{\varepsilon_g} \right)^{10/9} (1 + 3\chi)$$

$$\chi = \sqrt{\frac{(\sum \varepsilon_k \rho_k)(\sum \varepsilon_k \rho_k / d_k^2)}{(\sum \varepsilon_k \rho_k / d_k)^2} - 1} \quad k=l,s$$

$$k_k^* = 0.3289 \quad \phi = 7.26 \times 10^{-3}$$

A.3 Solid Viscosity from Kinetic Theory

A.3.1 Fluctuating Energy Equation. ($k=l,s$)

$$\frac{3}{2} \left[\frac{\partial}{\partial t} (\varepsilon_k \rho_k \Theta_k) + \nabla \cdot (\varepsilon_k \rho_k \Theta_k \mathbf{v}_k) \right] = \nabla \cdot (\kappa_k \nabla \Theta_k) - \gamma_k - 3\beta_{kg} \Theta_k + \Phi_k$$

A.3.2 Energy Production Rate

$$\Phi_k = [\tau_k] : \nabla \mathbf{v}_k \quad k=g,\ell,s$$

A.3.3 Conductive Coefficient of Fluctuating Energy

$$\kappa_k = \frac{2\kappa_{kas}}{(1+e_k)g_{0k}} \left\{ 1 + \frac{6}{5}(1+e_k)g_{0k}\varepsilon_k \right\}^2 + 2\varepsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

A.3.4 Collisional Energy Dissipation

$$\gamma_k = 3(1-e_k^2)\varepsilon_k^2 \rho_k g_{0k} \Theta_k \left\{ \frac{4\sqrt{\Theta_k}}{d_k \sqrt{\pi}} - \nabla \cdot \mathbf{v}_k \right\}$$

A.3.5 Solids Pressure

$$P_k = \rho_k \varepsilon_k \Theta_k \{ 1 + 2(1+e_k)g_{0k}\varepsilon_k \}$$

A.3.6 Solids Bulk Viscosity

$$\xi_k = \frac{4}{3} \varepsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

A.3.7 Solids Shear Viscosity

$$\mu_k = \frac{2\mu_{kas}}{(1+e_k)g_{0k}} \left\{ 1 + \frac{4}{5}(1+e_k)g_{0k}\varepsilon_k \right\}^2 + \frac{4}{5} \varepsilon_k^2 \rho_k d_k g_{0k} (1+e_k) \sqrt{\frac{\Theta_k}{\pi}}$$

Where

$$g_{0k} = \left\{ 1 - \left(\frac{\varepsilon_k}{\varepsilon_{k_{\max}}} \right)^{1/3} \right\}^{-1} \quad \mu_{k,d} = \frac{5}{96} \rho_k d_k \sqrt{\pi \Theta_k} \quad \kappa_{k,d} = \frac{75}{384} \rho_k d_k \sqrt{\pi \Theta_k}$$

A.4 Reaction and Mass Transfer

A.4.1 General Reactions

$$\sum_j^N \alpha_{ik}^j S^j = 0 \quad i=1,2,\dots,M_k \quad k=g,\ell,s$$

$$r_{ik} = f_{ik}(P_k, T_k, \varepsilon_k, y_k^1, y_k^2, \dots, y_k^N, \dots) \quad i=1,2,\dots,M_k \quad k=g,\ell,s$$

A.4.2 Mass Transfer Between Phases

$$R_k^j = f_k^j(P_k, T_k, a_k, y_k^1, y_k^2, \dots, y_k^N, P_g, T_g, y_g^1, y_g^2, \dots, y_g^N, \dots) \quad k=\ell,s$$

A.5 Species Balances ($j=1, 2, \dots, N$)

A.5.1 Gas Phase:

$$\frac{\partial}{\partial t}(\varepsilon_g \rho_g y_g^j) + \nabla \cdot (\varepsilon_g \rho_g y_g^j \mathbf{v}_g) = \sum_{i=1}^{M_g} \alpha_{ig}^j M^j r_{ig} - \sum_{k=\ell,s} M^j R_k^j$$

A.5.2 Liquid and Solid Phases: ($k=\ell,s$)

$$\frac{\partial}{\partial t}(\varepsilon_k \rho_k y_k^j) + \nabla \cdot (\varepsilon_k \rho_k y_k^j \mathbf{v}_k) = \sum_{i=1}^{M_k} \alpha_{ik}^j M^j r_{ik} + M^j R_k^j$$

$$\sum_j^N y_k^j = 1 \quad k=g,\ell,s$$

$$\text{A.5.3 Phase Changes: } (k=\ell,s) \quad \dot{m}_k = \sum_j^N M^j R_k^j \quad \dot{m}_g = - \sum_{k=\ell,s} \dot{m}_k$$

A.6 The Finite Difference Equations

The computations are carried out using a two-dimensional Eulerian mesh of non-uniform size finite-difference computational cells. These cells are rectangles with

dimensions dx_i (or dr_i) and dy_j (or dz_j). A typical mesh with cell flag (define the type of the cell) and computational cell (i, j) are shown in Figure A.1. The indexes i and j that label cell (i, j) count cell centers in the x-direction and y-direction, respectively, and assume only positive integer values. The scalar variables ($\epsilon_k, \rho_k, P_k, T_k, H_k, y_k^j, \dots$) are located at the cell center and the vector variables ($v_k, [\tau_k], \dots$) at the cell boundaries.

The finite-difference approximations to the hydrodynamic equations form a system of nonlinear algebraic equations relating quantities at time $t=(n+1)dt$, where n is zero or a positive integer and dt is the time increment by which these quantities advance each computational cycle.

A.6.1 Averaging Process. Quantities in the finite difference equations required at spatial locations other than where they are defined are obtained by weighted averaging.

(a) Cell Centered Quantities. The cell centered properties Ψ are defined at the cell center of (i, j) . At other locations averaging is used as follows,

$$\Psi_{i+\frac{1}{2},j} = \frac{1}{2\delta r_{i+\frac{1}{2}}} (\delta r_{i+1} \Psi_{i,j} + \delta r_i \Psi_{i+1,j})$$

$$\Psi_{i,j+\frac{1}{2}} = \frac{1}{2\delta z_{j+\frac{1}{2}}} (\delta z_{j+1} \Psi_{i,j} + \delta z_j \Psi_{i,j+1})$$

$$\begin{aligned} \Psi_{i+\frac{1}{2},j+\frac{1}{2}} = & \frac{1}{4\delta r_{i+\frac{1}{2}}\delta z_{j+\frac{1}{2}}} (\delta r_{i+1}\delta z_{j+1} \Psi_{i,j} + \delta r_i\delta z_{j+1} \Psi_{i+1,j} \\ & + \delta r_{i+1}\delta z_j \Psi_{i,j+1} + \delta r_i\delta z_j \Psi_{i+1,j+1}) \end{aligned}$$

(b) Boundary Centered Quantities. The boundary centered quantity in x-direction is u which defined at $(i, j+1/2)$. The averaging is as follows,

Figure A.1 The Computational Mesh

$$u_{i,j} = \frac{1}{2}(u_{i+\frac{1}{2},j} + u_{i+\frac{1}{2},j+1})$$

$$u_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2\delta z_{j+\frac{1}{2}}}(\delta z_{j+1}u_{i+\frac{1}{2},j} + \delta z_j u_{i+\frac{1}{2},j+1})$$

$$u_{i,j+\frac{1}{2}} = \frac{1}{4\delta z_{j+\frac{1}{2}}}(\delta z_{j+1}(u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j}) + \delta z_j(u_{i-\frac{1}{2},j+1} + u_{i+\frac{1}{2},j+1}))$$

The boundary centered quantity in y-direction is v which defined at $(i+1/2, j)$. The averaging is as follows,

$$v_{i,j} = \frac{1}{2}(v_{i+\frac{1}{2},j} + v_{i+\frac{1}{2},j+1})$$

$$v_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2\delta r_{i+\frac{1}{2}}}(\delta r_{i+1}u_{i,j+\frac{1}{2}} + \delta r_i v_{i+1,j+\frac{1}{2}})$$

$$v_{i+\frac{1}{2},j} = \frac{1}{4\delta r_{i+\frac{1}{2}}}(\delta r_{i+1}(v_{i,j-\frac{1}{2}} + v_{i,j+\frac{1}{2}}) + \delta r_i(v_{i+1,j-\frac{1}{2}} + v_{i+1,j+\frac{1}{2}}))$$

A.6.2 Continuity Equations. The continuity equations are differenced fully implicitly as follows, ($k=g,\ell,s$)

$$(\rho_k \varepsilon_k)_{i,j}^{n+1} = (\rho_k \varepsilon_k)_{i,j}^n - \frac{\delta t}{\delta r_i} \langle (\rho_k \varepsilon_k) u_k \rangle_{i,j}^{n+1} - \frac{\delta t}{\delta z_j} \langle (\rho_k \varepsilon_k) v_k \rangle_{i,j}^{n+1} + \delta t \cdot (\dot{m}_k)_{i,j}^n$$

The donor cell differencing aids computational stability without the introduction of explicit artificial viscosity.

A.6.3 Momentum Equations. The momentum equations are differenced over a staggered mesh (Figure A.2) using a scheme in which the convective terms are treated explicitly and all other terms are treated implicitly. the difference equations are,

Figure A.2 The Staggered Computational Mesh for Momentum equations

$$\begin{aligned}
& (\rho_k \varepsilon_k u_k)_{i+\frac{1}{2}j}^{n+1} = \overline{(\rho_k \varepsilon_k u_k)}_{i+\frac{1}{2}j} - \frac{\delta t}{\delta r_{i+\frac{1}{2}j}} \left((P_k)_{i+1,j}^{n+1} - (P_k)_{i,j}^{n+1} \right) \\
& - (w_k)_{i+\frac{1}{2}j}^{n+1} g_r \delta t + \delta t \sum_{m=g,\ell,s} (\beta_{mk})_{i+\frac{1}{2}j}^n \left((u_m)_{i+\frac{1}{2}j}^{n+1} - (u_k)_{i+\frac{1}{2}j}^{n+1} \right) \\
& + \frac{\delta t}{\delta r_{i+\frac{1}{2}j}} \left((\tau_{ck})_{i+1,j}^{n+1} - (\tau_{ck})_{i,j}^{n+1} \right) + \delta t \cdot (\dot{m}_k)_{i,j}^n (u_k)_{i+\frac{1}{2}j}^{n+1} \\
& (\rho_k \varepsilon_k v_k)_{i,j+\frac{1}{2}}^{n+1} = \overline{(\rho_k \varepsilon_k v_k)}_{i,j+\frac{1}{2}} - \frac{\delta t}{\delta z_{j+\frac{1}{2}}} \left((P_k)_{i,j+1}^{n+1} - (P_k)_{i,j}^{n+1} \right) \\
& - (w_k)_{i,j+\frac{1}{2}}^{n+1} g_z \delta t + \delta t \sum_{m=\ell,g,s} (\beta_{mk})_{i,j+\frac{1}{2}}^n \left((v_m)_{i,j+\frac{1}{2}}^{n+1} - (v_k)_{i,j+\frac{1}{2}}^{n+1} \right) \\
& + \frac{\delta t}{\delta r_{j+\frac{1}{2}}} \left((\tau_{ck})_{i,j+1}^{n+1} - (\tau_{ck})_{i,j}^{n+1} \right) + \delta t \cdot (\dot{m}_k)_{i,j}^n (v_k)_{i,j+\frac{1}{2}}^{n+1}
\end{aligned}$$

where for gas phase $w_g = \rho_g$ and $\tau_{cg} = 0$, and for particulate phases

$$w_k = \frac{\varepsilon_k}{\varepsilon_g} (\rho_k - \sum_{m=g,\ell,s} \varepsilon_m \rho_m) \quad k = \ell, s.$$

All the explicit terms are lumped into the “tilde” quantities as shown below,

$$\begin{aligned}
& \overline{(\rho_k \varepsilon_k u_k)}_{i+\frac{1}{2}j} = (\rho_k \varepsilon_k u_k)_{i+\frac{1}{2}j}^n - \frac{\delta t}{\delta r_{i+\frac{1}{2}j}} \langle (\rho_k \varepsilon_k u_k) u_k \rangle_{i+\frac{1}{2}j}^n - \frac{\delta t}{\delta z_j} \langle (\rho_k \varepsilon_k u_k) v_k \rangle_{i+\frac{1}{2}j}^n \\
& + \frac{\delta t}{\delta r_{i+\frac{1}{2}j}} \left((\tau_{krr})_{i+1,j}^n - (\tau_{krr})_{i,j}^n \right) + \frac{\delta t}{\delta z_j} \left((\tau_{krz})_{i+\frac{1}{2}j+\frac{1}{2}}^n - (\tau_{krz})_{i+\frac{1}{2}j-\frac{1}{2}}^n \right) \\
& \overline{(\rho_k \varepsilon_k v_k)}_{i,j+\frac{1}{2}} = (\rho_k \varepsilon_k v_k)_{i,j+\frac{1}{2}}^n - \frac{\delta t}{\delta r_i} \langle (\rho_k \varepsilon_k v_k) u_k \rangle_{i,j+\frac{1}{2}}^n - \frac{\delta t}{\delta z_{j+\frac{1}{2}}} \langle (\rho_k \varepsilon_k v_k) v_k \rangle_{i,j+\frac{1}{2}}^n \\
& + \frac{\delta t}{\delta r_i} \left((\tau_{kzz})_{i+\frac{1}{2}j+\frac{1}{2}}^n - (\tau_{kzz})_{i-\frac{1}{2}j+\frac{1}{2}}^n \right) + \frac{\delta t}{\delta z_{j+\frac{1}{2}}} \left((\tau_{kzz})_{i+j+1}^n - (\tau_{kzz})_{i,j}^n \right)
\end{aligned}$$

A.6.4 Energy Equations

(a) Gas Phase.

$$(\rho_g \varepsilon_g \mathbf{H}_g)_{i,j}^{n+1} = (\rho_g \varepsilon_g \mathbf{H}_g)_{i,j}^n + \overline{(\rho_g \varepsilon_g \mathbf{H}_g)_{i,j}} + \delta t \sum_{k=\ell,s} (\mathbf{h}_{vk})_{i,j}^n (T_k - T_g)_{i,j}^{n+1} \\ + \delta t \sum_{k=\ell,s} (\beta_{kg})_{i,j}^n \left\{ \left((\mathbf{u}_k)_{i,j}^n - (\mathbf{u}_g)_{i,j}^n \right)^2 + \left((\mathbf{v}_k)_{i,j}^n - (\mathbf{v}_g)_{i,j}^n \right)^2 \right\} + \delta t \cdot (\dot{Q}_{rg} + \dot{Q}_{fg})_{i,j}^n$$

(b) Solid Phases.

$$(\rho_k \varepsilon_k \mathbf{H}_k)_{i,j}^{n+1} = (\rho_k \varepsilon_k \mathbf{H}_k)_{i,j}^n + \overline{(\rho_k \varepsilon_k \mathbf{H}_k)_{i,j}} + \delta t (\mathbf{h}_{vk})_{i,j}^n (T_g - T_k)_{i,j}^{n+1} + \delta t \cdot (\dot{Q}_{rk} + \dot{Q}_{fk})_{i,j}^n$$

(c) "Tilde" quantities are given by,

$$\overline{(\rho_g \varepsilon_g \mathbf{H}_g)_{i,j}} = -\frac{\delta t}{\delta r_i} \langle (\rho_g \varepsilon_g)^{n+1} \overline{\mathbf{H}}_g (\mathbf{u}_g)^{n+1} \rangle_{i,j} - \frac{\delta t}{\delta z_j} \langle (\rho_g \varepsilon_g)^{n+1} \overline{\mathbf{H}}_g (\mathbf{u}_g)^{n+1} \rangle_{i,j} \\ + \left((\mathbf{P}_g)_{i,j}^{n+1} - (\mathbf{P}_g)_{i,j}^n \right) + \frac{\delta t}{\delta r_i} (\mathbf{u}_g)_{i,j}^{n+1} \left((\mathbf{P}_g)_{i+\frac{1}{2},j}^{n+1} - (\mathbf{P}_g)_{i-\frac{1}{2},j}^{n+1} \right) \\ - \frac{\delta t}{\delta z_j} (\mathbf{v}_g)_{i,j}^{n+1} \left((\mathbf{P}_g)_{i,j+\frac{1}{2}}^{n+1} - (\mathbf{P}_g)_{i,j-\frac{1}{2}}^{n+1} \right) \\ \overline{(\varepsilon_k \rho_k \mathbf{H}_k)_{i,j}} = -\frac{\delta t}{\delta r_i} \langle (\varepsilon_k \rho_k)^{n+1} \overline{\mathbf{H}}_k (\mathbf{u}_k)^{n+1} \rangle_{i,j} - \frac{\delta t}{\delta z_j} \langle (\varepsilon_k \rho_k)^{n+1} \overline{\mathbf{H}}_k (\mathbf{v}_k)^{n+1} \rangle_{i,j} \\ + \frac{\delta t}{\delta r_i} \left((\mathbf{K}_k)_{i+\frac{1}{2},j}^n \frac{(\overline{\mathbf{T}}_k)_{i+1,j} - (\overline{\mathbf{T}}_k)_{i,j}}{\delta r_{i+\frac{1}{2}}} - (\mathbf{K}_k)_{i-\frac{1}{2},j}^n \frac{(\overline{\mathbf{T}}_k)_{i,j} - (\overline{\mathbf{T}}_k)_{i-1,j}}{\delta r_{i-\frac{1}{2}}} \right) \\ + \frac{\delta t}{\delta z_j} \left((\mathbf{K}_k)_{i,j+\frac{1}{2}}^n \frac{(\overline{\mathbf{T}}_k)_{i,j+1} - (\overline{\mathbf{T}}_k)_{i,j}}{\delta z_{i+\frac{1}{2}}} - (\mathbf{K}_k)_{i,j-\frac{1}{2}}^n \frac{(\overline{\mathbf{T}}_k)_{i,j} - (\overline{\mathbf{T}}_k)_{i,j-1}}{\delta z_{i-\frac{1}{2}}} \right) + \delta t (\Phi_k)_{i,j}^n \\ (\Phi_k)_{i,j}^n = (\tau_{kr})_{i,j}^n \frac{(\mathbf{u}_k)_{i+\frac{1}{2},j}^{n+1} - (\mathbf{u}_k)_{i-\frac{1}{2},j}^{n+1}}{\delta r_i} + (\tau_{kz})_{i,j}^n \frac{(\mathbf{v}_k)_{i,j+\frac{1}{2}}^{n+1} - (\mathbf{v}_k)_{i,j-\frac{1}{2}}^{n+1}}{\delta z_j} + \frac{((\tau_{kzz})_{i,j}^n)^2}{2(\mu_k)_{i,j}^n}$$

$$(\dot{Q}_{rg})_{i,j}^n = \sum_{ix=1}^{ixn} (r_{ixg})_{i,j}^n (\Delta H_{ixg})_{i,j}^n - \sum_{m=\ell,s} (\dot{m}_m)_{i,j}^n (H_g)_{i,j}^n$$

$$(\dot{Q}_{fg})_{i,j}^n = \begin{cases} 0 & \text{if cell (i,j) has no contact} \\ & \text{with a heat boundary or block} \\ (C_{fx} \epsilon_g)_{i,j}^n & \text{if cell (i,j) contacts a heat boundary} \\ & \text{or block with constant heat flux} \\ (K_B \epsilon_g (T_B - T_g))_{i,j}^n & \text{if cell (i,j) contacts a heat boundary} \\ & \text{or block with constant temperature} \end{cases}$$

$$(\dot{Q}_{rk})_{i,j}^n = (\dot{m}_k)_{i,j}^n (H_g)_{i,j}^n + \sum_{ix=1}^{ixn} (r_{ixk})_{i,j}^n (\Delta H_{ixk})_{i,j}^n$$

$$(\dot{Q}_{rk})_{i,j}^n = \begin{cases} 0 & \text{if cell (i,j) has no contact} \\ & \text{with a heat boundary or block} \\ (C_r \epsilon_k)_{i,j}^n & \text{if cell (i,j) contacts a heat boundary} \\ & \text{or block with constant heat flux} \\ (K_B \epsilon_k (T_B - T_k))_{i,j}^n & \text{if cell (i,j) contacts a heat boundary} \\ & \text{or block with constant temperature} \end{cases}$$

A.6.5 Species Balance. (k=g,ℓ,s), j=the j-th species.

$$(\rho_k \epsilon_k y_k^j)_{i,j}^{n+1} = (\rho_k \epsilon_k y_k^j)_{i,j}^n - \frac{\delta t}{\delta r_i} \langle (\rho_k \epsilon_k y_k^j) u_k \rangle_{i,j}^{n+1} - \frac{\delta t}{\delta z_j} \langle (\rho_k \epsilon_k y_k^j) v_k \rangle_{i,j}^{n+1} + \delta t \cdot (\dot{m}_k)_{i,j}^n$$

As mentioned before, the flux quantities denoted by $\langle \psi \mathbf{u}_k \rangle$ and $\langle \psi \mathbf{v}_k \rangle$ are calculated using donor-cell differencing, where ψ refers to quantities:

$$\varepsilon_k \rho_k, \varepsilon_k \rho_k \mathbf{u}_k, \varepsilon_k \rho_k \mathbf{v}_k, \varepsilon_k \rho_k \bar{H}_k, \varepsilon_k \rho_k y_k^j \mathbf{u}_k, \varepsilon_k \rho_k y_k^j \mathbf{v}_k \quad k=g,\ell,s$$

The angular brackets represent donor cell differenced quantities as shown below,

$$\langle \Psi \mathbf{u}_k \rangle_{m,p} = (\mathbf{u}_k)_{m+\frac{1}{2},p} \begin{cases} (\Psi)_{m,p} & \text{if } (\mathbf{u}_k)_{m+\frac{1}{2},p} \geq 0 \\ (\Psi)_{m+1,p} & \text{if } (\mathbf{u}_k)_{m+\frac{1}{2},p} < 0 \end{cases} - (\mathbf{u}_k)_{m-\frac{1}{2},p} \begin{cases} (\Psi)_{m-1,p} & \text{if } (\mathbf{u}_k)_{m-\frac{1}{2},p} \geq 0 \\ (\Psi)_{m,p} & \text{if } (\mathbf{u}_k)_{m-\frac{1}{2},p} < 0 \end{cases}$$

$$\langle \Psi \mathbf{v}_k \rangle_{m,p} = (\mathbf{v}_k)_{m,p+\frac{1}{2}} \begin{cases} (\Psi)_{m,p} & \text{if } (\mathbf{v}_k)_{m,p+\frac{1}{2}} \geq 0 \\ (\Psi)_{m,p+1} & \text{if } (\mathbf{v}_k)_{m,p+\frac{1}{2}} < 0 \end{cases} - (\mathbf{v}_k)_{m,p-\frac{1}{2}} \begin{cases} (\Psi)_{m,p-1} & \text{if } (\mathbf{v}_k)_{m,p-\frac{1}{2}} \geq 0 \\ (\Psi)_{m,p} & \text{if } (\mathbf{v}_k)_{m,p-\frac{1}{2}} < 0 \end{cases}$$

The viscous stress components are calculated with standard centered differencing,

i.e.,

$$(\nabla \cdot \mathbf{V}_k)_{i,j} = \frac{(\mathbf{u}_k)_{i+\frac{1}{2},j} - (\mathbf{u}_k)_{i-\frac{1}{2},j}}{\delta r_i} + \frac{(\mathbf{v}_k)_{i,j+\frac{1}{2}} - (\mathbf{v}_k)_{i,j-\frac{1}{2}}}{\delta z_j}$$

$$(\tau_{krr})_{i,j} = 2(\mu_k)_{i,j} \left(\frac{(u_k)_{i+\frac{1}{2},j} - (u_k)_{i-\frac{1}{2},j}}{\delta r_i} \right) + (\xi_k - \frac{2}{3}\mu_k)_{i,j} (\nabla \cdot \mathbf{V}_k)_{i,j}$$

$$(\tau_{kzz})_{i,j} = 2(\mu_k)_{i,j} \left(\frac{(v_k)_{i,j+\frac{1}{2}} - (v_k)_{i,j-\frac{1}{2}}}{\delta z_j} \right) + (\xi_k - \frac{2}{3}\mu_k)_{i,j} (\nabla \cdot \mathbf{V}_k)_{i,j}$$

$$(\tau_{krz})_{i,j} = 2(\mu_k)_{i,j} \left(\frac{(u_k)_{i,j+1} - (u_k)_{i,j-1}}{\delta z_{j-\frac{1}{2}} + \delta z_{j+\frac{1}{2}}} + \frac{(v_k)_{i+1,j} - (v_k)_{i-1,j}}{\delta r_{i-\frac{1}{2}} + \delta r_{i+\frac{1}{2}}} \right)$$

$$(\tau_{krz})_{i+\frac{1}{2},j+\frac{1}{2}} = 2(\mu_k)_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{(u_k)_{i+\frac{1}{2},j+1} - (u_k)_{i+\frac{1}{2},j}}{\delta z_{j+\frac{1}{2}}} + \frac{(v_k)_{i+1,j+\frac{1}{2}} - (v_k)_{i,j-\frac{1}{2}}}{\delta r_{i+\frac{1}{2}}} \right)$$

A.6.6. Fluctuating Energy Equation

$$\begin{aligned} \frac{2}{3}(\varepsilon_k \rho_k \Theta_k)_{i,j}^{n+1} &= \frac{2}{3}(\varepsilon_k \rho_k \Theta_k)_{i,j}^n + \overline{(\varepsilon_k \rho_k \Theta_k)}_{i,j} - \delta t (\mathbf{P}_k)_{i,j}^{n+1} (\nabla \cdot \mathbf{V}_k)_{i,j}^{n+1} \\ &+ \delta t (\Phi_k)_{i,j}^{n+1} - \delta t (\gamma_k)_{i,j}^{n+1} - 3\delta t (\beta_{gk})_{i,j}^n (\Theta_k)_{i,j}^{n+1} \end{aligned}$$

$$\overline{(\varepsilon_k \rho_k \Theta_k)}_{i,j} = \frac{2}{3} \frac{\delta t}{\delta r_i} \langle (\varepsilon_k \rho_k)^{n+1} (\Theta_k)^n (u_k)^{n+1} \rangle_{i,j}$$

$$\frac{2}{3} \frac{\delta t}{\delta z_j} \langle (\varepsilon_k \rho_k)^{n+1} (\Theta_k)^n (v_k)^{n+1} \rangle_{i,j}$$

$$+ \frac{\delta t}{\delta r_i} \left((\kappa_k)_{i+\frac{1}{2},j}^n \frac{(\Theta_k)_{i+1,j} - (\Theta_k)_{i,j}}{\delta r_{i+\frac{1}{2}}} - (\kappa_k)_{i-\frac{1}{2},j}^n \frac{(\Theta_k)_{i,j} - (\Theta_k)_{i-1,j}}{\delta r_{i-\frac{1}{2}}} \right)$$

$$+ \frac{\delta t}{\delta z_j} \left((\kappa_k)_{i,j+\frac{1}{2}}^n \frac{(\Theta_k)_{i,j+1} - (\Theta_k)_{i,j}}{\delta z_{i+\frac{1}{2}}} - (\kappa_k)_{i,j-\frac{1}{2}}^n \frac{(\Theta_k)_{i,j} - (\Theta_k)_{i,j-1}}{\delta z_{i-\frac{1}{2}}} \right)$$

A.7 Solution Technique for Finite -Difference Equations

An iterative technique is used to solve the difference equations given in the previous section. The solution technique for finite -difference equations may be applied to any multiple particulate phases, not limited liquid and solid.

A.7.1 Solution of the Momentum Equations. To facilitate the particular method of solution the equations are recast in the following form. The momentum equations in x-direction could be collected together in a matrix form.

$$(A)_{i+\frac{1}{2},j} (U)_{i+\frac{1}{2},j}^n = (B_u)_{i+\frac{1}{2},j}$$

$$A = \begin{pmatrix} A_{gg} & A_{g1} & A_{g2} & \cdots & A_{gN} \\ A_{1g} & A_{11} & A_{12} & \cdots & A_{1N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{Ng} & A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix}$$

where

$$A_{kk} = (\epsilon_k \rho_k)^{n+1} + \delta t \sum_{\substack{\ell=g,1 \\ \ell \neq k}}^N (\beta_{\ell k})^n - \delta t (\dot{m}_k)^n \quad k=g,1,2,\dots,N$$

$$A_{km} = A_{mk} = -\delta t (\beta_{mk})^n \quad k,m=g,1,2,\dots,N \quad m \neq k$$

$$A_{i+\frac{1}{2},j} \cdot \begin{pmatrix} u_g \\ u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} (\overline{\epsilon_g \rho_g u_g}) - \frac{\delta t}{\delta r_{i+\frac{1}{2}}} \left((P_g + \tau_{cg})_{i+1,j}^{n+1} - (P_g + \tau_{cg})_{i,j}^{n+1} \right) - (w_g) g_r \delta t \\ (\overline{\epsilon_1 \rho_1 u_1}) - \frac{\delta t}{\delta r_{i+\frac{1}{2}}} \left((P_1 + \tau_{c1})_{i+1,j}^{n+1} - (P_1 + \tau_{c1})_{i,j}^{n+1} \right) - (w_1) g_r \delta t \\ \vdots \\ (\overline{\epsilon_N \rho_N u_N}) - \frac{\delta t}{\delta r_{i+\frac{1}{2}}} \left((P_N + \tau_{cN})_{i+1,j}^{n+1} - (P_N + \tau_{cN})_{i,j}^{n+1} \right) - (w_N) g_r \delta t \end{pmatrix}$$

where for gas phase $w_g = \rho_g$ and $\tau_{cg} = 0$, and for particulate phases

$$w_k = \frac{\epsilon_k}{\epsilon_g} (\rho_k - \sum_{m=g,1}^N \epsilon_m \rho_m) \quad k=1,2,\dots,N.$$

and similarly, momentum equation in y-direction can be written as,

$$(A)_{i,j+\frac{1}{2}} (V)_{i,j+\frac{1}{2}}^n = (B_v)_{i,j+\frac{1}{2}} \\ A_{i,j+\frac{1}{2}} \begin{pmatrix} v_g \\ v_1 \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} (\overline{\epsilon_g \rho_g v_g}) - \frac{\delta t}{\delta z_{j+\frac{1}{2}}} ((P_g + \tau_{cg})_{i,j+1}^{n+1} - (P_g + \tau_{cg})_{i,j}^{n+1}) - (w_g) g_z \delta t \\ (\overline{\epsilon_1 \rho_1 v_1}) - \frac{\delta t}{\delta z_{j+\frac{1}{2}}} ((P_1 + \tau_{c1})_{i,j+1}^{n+1} - (P_1 + \tau_{c1})_{i,j}^{n+1}) - (w_k) g_z \delta t \\ \vdots \\ (\overline{\epsilon_N \rho_N v_N}) - \frac{\delta t}{\delta z_{j+\frac{1}{2}}} ((P_N + \tau_{cN})_{i,j+1}^{n+1} - (P_N + \tau_{cN})_{i,j}^{n+1}) - (w_N) g_z \delta t \end{pmatrix}$$

A.7.2 Convergence on Fluid Continuity Equation. The solution process is carried out in three major steps. First of all, the continuity equations, the momentum equations, and a part of energy equation are solved simultaneously to establish the pressure and the velocity fields. In this step only the interface heat transfer part of the energy equation is considered. Secondly, the remaining parts of the energy equations are solved to establish the temperature profiles. Finally, the compositions for each phase are solved based on the pressure and the velocity fields, volume fraction and temperature profiles obtained from the previous steps. The solution procedure of computational sweep is illustrated in Figure A.3.

The first step proceeds as follows: The calculations are started with a guessed pressure field that is either the specified initial condition or the pressure field computed in the previous time

Figure A.3 The Computational Sweep

step. Using this guessed pressure field, the velocities are calculated from the matrices above. The particulate phase continuity equations are solved using the updated velocities to calculate the particulate phase volume fractions. The gas volume fraction, ϵ_g , is then calculated from the following equation,

$$\epsilon_g = 1 - \sum_{k=1}^N \epsilon_k$$

Using ϵ_g and the updated fluid velocities, the residue of the fluid continuity equations, $D_{i,j}$, is calculated,

$$D_{i,j} = -(\epsilon_g \rho_g)_{i,j}^{n+1} + (\epsilon_g \rho_g)_{i,j}^n - \frac{\delta t}{\delta r_i} \langle (\epsilon_g \rho_g) u_g \rangle_{i,j}^{n+1} - \frac{\delta t}{\delta z_j} \langle (\epsilon_g \rho_g) v_g \rangle_{i,j}^{n+1} + \delta t (\dot{m}_g)_g^n$$

Ideally, for a converged solution, $D_{i,j}$ should be zero. In the code, $D_{i,j}$ is compared with a very small number. The value of the convergence criterion is,

$$D_{i,j} \leq \text{CONV}_{i,j}^{n+1} = \text{EPSG} \cdot (\epsilon_g \rho_g)_{i,j}^n$$

where EPSG is read in. The default and recommended value of EPSG is 10^{-5} .

The computations begin at the left-bottom corner fluid cell. The pressure is corrected in one cell at a time until convergence is obtained or the number of iterations exceed an inner iteration limit. The computations proceed from left to right and from bottom to top until the entire computational regime is covered. At the end of such a computational sweep, if a pressure adjustment was necessary in any of the cells, the procedure is repeated until simultaneous convergence in all the cells is obtained. The number of iterations, however, is restricted by an outer iteration limit.

A.7.3 Pressure Iteration. When $D_{i,j}$ fails to meet the convergence criterion in any cell, the pressure is adjusted using a combination of Newton's method and secant method. The initial adjustment of pressure uses Newton's method.

$$(\mathbf{P}_g)^{m+1} = (\mathbf{P}_g)^m - \omega \frac{\mathbf{D}^m}{\partial \mathbf{D} / \partial (\mathbf{P}_g)^m}$$

where the indices i, j , and n have been omitted. The index, m , indicates the iteration level. This is equivalent to using Newton's method for each cell, where ω is a relaxation parameter near unity, and $\bar{\beta}$ is computed as,

$$\frac{1}{\bar{\beta}_{i,j}} = \frac{\partial \mathbf{D}_{i,j}}{\partial (\mathbf{P}_g)_{i,j}} = \frac{\varepsilon_g}{C_{i,j}^2} + \left(\frac{\delta t}{\delta r_i} \right)^2 \left((\varepsilon_g)_{i+\frac{1}{2},j} + (\varepsilon_g)_{i-\frac{1}{2},j} \right) + \left(\frac{\delta t}{\delta z_j} \right)^2 \left((\varepsilon_g)_{i,j+\frac{1}{2}} + (\varepsilon_g)_{i,j-\frac{1}{2}} \right)$$

once every time step. The sound speed $C_{i,j}$ is given by,

$$C_{i,j}^2 = \left(\frac{\partial \mathbf{P}_g}{\partial \rho_g} \right)_{i,j}$$

where $(\partial \mathbf{P}_g / \partial \rho_g)$ can be determined from the equation of state.

This formulation is only approximate. Hence, subsequent corrections use the secant method:

$$(\mathbf{P}_g)^{m+1} = (\mathbf{P}_g)^m - \omega \left(\frac{(\mathbf{P}_g)^{m-1} - (\mathbf{P}_g)^m}{\mathbf{D}^{m-1} - \mathbf{D}^m} \right) \mathbf{D}^m$$

The use of secant method is continued until $\mathbf{D}_{i,j}$ changes sign. Thereafter a combination of the secant method and a bisection method is used. The method is illustrated in the Figure A.4.

Given the three pressures P_1 , P_2 , and P_3 of which P_1 and P_2 bracket the desired pressure and P_3 lies between them and the respective mass residuals D_1 , D_2 , and D_3 , do not satisfy the convergence criterion in cell $D_1 > 0$, and $D_2 < 0$. With three pressures and their mass residuals obtained as described, or otherwise a constrained two-sided secant technique is used to obtain further pressure adjustments. From these pressures and their

Figure A.4 The Secant method for Pressure Iteration

mass residuals, the pressure P_A and P_B are determined by straight line extrapolation and interpolation, respectively, as follows,

$$p_A = \begin{cases} (p_3 D_1 - p_1 D_3) / (D_1 - D_3) & \text{for } D_1 \neq D_3 \\ (p_2 + p_3) / 2 & \text{for } D_1 = D_3 \end{cases}$$

$$p_B = \begin{cases} (p_3 D_2 - p_2 D_3) / (D_2 - D_3) & \text{for } D_2 \neq D_3 \\ (p_1 + p_3) / 2 & \text{for } D_2 = D_3 \end{cases}$$

The new estimate of the advanced time pressure is then computed as,

$$(P_g)^{m+1} = \frac{1}{2}(p_A + p_B)$$

If the pressure, P_A should lie outside the interval P_1 to P_3 , it is given the value $(P_A + P_B)/2$. After $(P_g)^{m+1}$ is estimated, point 2 is discarded and points 1 and 3 are retained as improved bounds for the next pressure estimate. When $D_{i,j}$ changes sign, the value of $\bar{\beta}_{i,j}$ is also updated for future iterations as,

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

A.7.4 Solution of the Energy Equations. The specific enthalpies $\bar{H}_k$ are calculated in subroutine IGIL accounting for the mass, momentum and energy exchange rates. For the iterative part of the solution, a simplified set of energy equations is used, which is differenced as follows,

$$(\epsilon_g \rho_g)_{i,j}^{n+1} (\bar{H}_g)_{i,j} = (\epsilon_g \rho_g)_{i,j}^{n+1} (\bar{H}_g)_{i,j}^n + \frac{\delta t}{2} \sum_{k=1}^N (h_{vk})_{i,j}^n (\bar{T}_k - \bar{T}_g)$$

$$+ \delta t \sum_{k=1}^N (\beta_{kg})_{i,j}^n \left\{ \left[(u_k)_{i,j}^n - (u_g)_{i,j}^n \right]^2 + \left[(v_k)_{i,j}^n - (v_g)_{i,j}^n \right]^2 \right\} + (\dot{Q}_{rg} + \dot{Q}_{fg})_{i,j}^n$$

$$(\epsilon_k \rho_k)_{i,j}^{n+1} (\bar{H}_k)_{i,j} = (\epsilon_k \rho_k)_{i,j}^{n+1} (\bar{H}_k)_{i,j}^n + \frac{\delta t}{2} (h_{vk})_{i,j}^n (\bar{T}_g - \bar{T}_k) + (\dot{Q}_{rk} + \dot{Q}_{fk})_{i,j}^n$$

Note that only half the effect of the interface heat transfer is considered here. Thus $\bar{H}$ is some intermediate value between H^n and H^{n+1} defined as,

$$\bar{H}_k = H_k^n + (\bar{T}_k - T_k^n) C_k$$

where, C_k is the specific heat of the fluid or the particulate phases. Rearranging, we get,

$$\bar{T}_k = \frac{\bar{H}_k - H_k^n}{C_k} + T_k^n$$

Thus we get,

$$\begin{aligned} (\varepsilon_g \rho_g)_{i,j}^{n+1} (\bar{H}_g)_{i,j} &= (\varepsilon_g \rho_g)_{i,j}^{n+1} (H_g)_{i,j}^n + (\dot{Q}_{rg} + \dot{Q}_{fg})_{i,j}^n + \\ &\frac{\delta t}{2} \sum_{k=1}^N (h_{vk})_{i,j}^n \left(\frac{(\bar{H}_k)_{i,j} - (H_k)_{i,j}^n}{C_k} + (T_k)_{i,j}^n - \frac{(\bar{H}_g)_{i,j} - (H_g)_{i,j}^n}{C_g} - (T_g)_{i,j}^n \right) \\ &+ \delta t \sum_{k=1}^N (\beta_{kg})_{i,j}^n \left\{ \left((u_k)_{i,j}^n - (u_g)_{i,j}^n \right)^2 + \left((v_k)_{i,j}^n - (v_g)_{i,j}^n \right)^2 \right\} \\ (\varepsilon_k \rho_k)_{i,j}^{n+1} (\bar{H}_k)_{i,j} &= (\varepsilon_k \rho_k)_{i,j}^{n+1} (H_k)_{i,j}^n + (\dot{Q}_{rk} + \dot{Q}_{fk})_{i,j}^n \\ &+ \frac{\delta t}{2} (h_{vk})_{i,j}^n \left(\frac{(\bar{H}_g)_{i,j} - (H_g)_{i,j}^n}{C_g} + (T_g)_{i,j}^n - \frac{(\bar{H}_k)_{i,j} - (H_k)_{i,j}^n}{C_k} - (T_k)_{i,j}^n \right) \end{aligned}$$

In matrix form,

$$(A_h)_{i,j} (\bar{H})_{i,j} = (B_h)_{i,j}$$

Note that $(\varepsilon_k \rho_k)$, $k=g, \ell, s$ are evaluated at time $(n+1)$ and H_k , T_k , h_{vk} are evaluated at time n .

$$(A_h)_{i,j} = \begin{pmatrix} \varepsilon_g \rho_g + \frac{\delta t}{2C_g} \sum_{k=1}^N h_{vk} & -\frac{\delta t}{2C_1} h_{v1} & -\frac{\delta t}{2C_2} h_{v2} & \cdots & -\frac{\delta t}{2C_N} h_{vN} \\ -\frac{\delta t}{2C_g} h_{v1} & \varepsilon_1 \rho_1 + \frac{\delta t}{2C_1} h_{v1} & 0 & \cdots & 0 \\ -\frac{\delta t}{2C_g} h_{v2} & 0 & \varepsilon_2 \rho_2 + \frac{\delta t}{2C_2} h_{v2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\delta t}{2C_g} h_{vN} & 0 & 0 & \cdots & \varepsilon_N \rho_N + \frac{\delta t}{2C_N} h_{vN} \end{pmatrix}_{i,j}$$

$$(\bar{\mathbf{H}})_{i,j} = \begin{pmatrix} \bar{\mathbf{H}}_g \\ \bar{\mathbf{H}}_1 \\ \bar{\mathbf{H}}_2 \\ \vdots \\ \bar{\mathbf{H}}_N \end{pmatrix}_{i,j}$$

$$(\mathbf{B}_h)_{i,j} = \begin{pmatrix} (\epsilon_g \rho_g) \mathbf{H}_g + (\dot{\mathbf{Q}}_{rg} + \dot{\mathbf{Q}}_{fg}) + \frac{\delta t}{2} \sum_{k=1}^N \left\{ \mathbf{h}_{vk} \left(\frac{\mathbf{H}_g}{C_g} - \frac{\mathbf{H}_k}{C_k} + \mathbf{T}_k - \mathbf{T}_g \right) + \beta_{kg} ((\mathbf{u}_k - \mathbf{u}_g)^2 + (\mathbf{v}_k - \mathbf{v}_g)^2) \right\} \\ (\epsilon_1 \rho_1) \mathbf{H}_1 + \frac{\delta t}{2} \mathbf{h}_{v1} \left(\frac{\mathbf{H}_1}{C_1} - \frac{\mathbf{H}_g}{C_g} + \mathbf{T}_g - \mathbf{T}_1 \right) \\ (\epsilon_2 \rho_2) \mathbf{H}_2 + \frac{\delta t}{2} \mathbf{h}_{v2} \left(\frac{\mathbf{H}_2}{C_2} - \frac{\mathbf{H}_g}{C_g} + \mathbf{T}_g - \mathbf{T}_2 \right) \\ \vdots \\ (\epsilon_N \rho_N) \mathbf{H}_N + \frac{\delta t}{2} \mathbf{h}_{vN} \left(\frac{\mathbf{H}_N}{C_N} - \frac{\mathbf{H}_g}{C_g} + \mathbf{T}_g - \mathbf{T}_N \right) \end{pmatrix}_{i,j}$$

After the converged solution is obtained for the continuity and momentum equations, the solution of the energy equations is completed. The part of the energy equation solved during the iterative solution of the momentum and continuity equations is subtracted from the complete energy equations. Assuming that

$$(\bar{\mathbf{T}}_g - \bar{\mathbf{T}}_k) \approx (\mathbf{T}_g - \mathbf{T}_k)^{n+1}$$

we get,

$$\mathbf{T}_k^{n+1} = \frac{\mathbf{H}_k^{n+1} - \bar{\mathbf{H}}_k}{C_k} + \bar{\mathbf{T}}_k$$

$$\begin{aligned}
(\epsilon_g \rho_g)^{n+1} (\mathbf{H}_g)_{i,j} &= (\epsilon_g \rho_g)_{i,j}^{n+1} ((\bar{\mathbf{H}}_g)_{i,j} - (\mathbf{H}_g)_{i,j}^n) + (\epsilon_g \rho_g)_{i,j}^n (\mathbf{H}_g)_{i,j}^n + \overline{(\epsilon_g \rho_g \mathbf{H}_g)}_{i,j} \\
&+ \frac{\delta t}{2} \sum_{k=1}^N (\mathbf{h}_{vk})_{i,j}^n \left(\frac{(\mathbf{H}_k)_{i,j}^{n+1} - (\bar{\mathbf{H}}_k)_{i,j}}{C_k} + (\bar{\mathbf{T}}_k)_{i,j} - \frac{(\mathbf{H}_g)_{i,j}^{n+1} - (\bar{\mathbf{H}}_g)_{i,j}}{C_g} - (\bar{\mathbf{T}}_g)_{i,j} \right) + (\dot{\mathbf{Q}}_{rg} + \dot{\mathbf{Q}}_{fg})_{i,j}^n
\end{aligned}$$

$$\begin{aligned}
(\epsilon_k \rho_k)_{i,j}^{n+1} (H_k)_{i,j}^{n+1} &= (\epsilon_k \rho_k)_{i,j}^{n+1} ((\bar{H}_k)_{i,j} - (H_k)_{i,j}^n) + (\epsilon_k \rho_k)_{i,j}^n (H_k)_{i,j}^n + (\overline{\epsilon_k \rho_k H_k})_{i,j} \\
+ \frac{\delta t}{2} (h_{vk})_{i,j}^n &\left(\frac{(H_g)_{i,j}^{n+1} - (\bar{H}_g)_{i,j}}{C_g} + (\bar{T}_g)_{i,j} - \frac{(H_k)_{i,j}^{n+1} - (\bar{H}_k)_{i,j}}{C_k} - (\bar{T}_k)_{i,j} \right) + (\dot{Q}_{rk} + \dot{Q}_{fk})_{i,j}^n
\end{aligned}$$

Solution procedure is same as the one discussed above.

A.8 Boundary Conditions

The rectangular region in which calculations are to be performed is partitioned into cells of sizes dx_i (or dr_i) in x (or r)-direction and dy_i (or dz_i) in y (or z)-direction. A perimeter of fictitious (dummy) boundary cells surrounding the computing mesh is used to enforce boundary conditions. Several boundary conditions around the computing mesh perimeter are programmed in the code. The cell flag types are indicated by IFL(I, J).

Flag Cell Type

1	Fluid cell
2	No-slip rigid walls
3	Free-slip rigid walls for gas/liquid phases, and Partial-slip rigid walls for solid phases
4	Continuous outflow,
5	Prescribed inflow rate,
6	Prescribed pressure inflow,
7	Prescribed pressure outflow with particulate outflow,
8	Prescribed pressure outflow with no particulate outflow,

In the bottom row and left column of boundary cells, any number of inflow openings can be specified using flag types 5 and 6. Similarly, in the top row and right column of boundary cells, any number of outflow openings can be specified using flag

types 4, 7 and 8. Flag types 2 and 3 may be prescribed on any of the four boundaries to represent rigid (solid) cells. Obstacles blocks within the computing mesh are built from rigid cells, flag types (IFL = 2 or 3).

The position of all blocks must coincide with rectangular cells within the computing mesh. Calculations are not performed in the obstacle cells, only in the remaining fluid cells within the computing mesh. Cell flag type (IFL = 1) is a computational cell.

A.8.1 Rigid Cells. Three types of boundary conditions may be specified for a rigid (solid) cell: free-slip, no-slip or partial slip (IFL = 2 or 3). In two dimensions, a free-slip boundary represents line of symmetry and a non-adhering boundary that exerts no drag on the fluid; a no-slip boundary represents a viscous boundary that exerts a drag on the fluid.

Consider cell (i, j), which is a fluid cell, an inflow boundary cell, or an outflow boundary cell. An adiabatic rigid cell is said to be a corner cell if and only if it has at least two adjacent edges, each of which is shared with a fluid cell, an inflow boundary cell, or an outflow boundary cell.

If right cell (i+1, j) or left cell (i-1, j), is a rigid cell, then for all time levels n

$$\begin{aligned} (u_k)_{i+\frac{1}{2},j}^{n+1} &= 0 \\ (v_k)_{i,j+\frac{1}{2}}^{n+1} &= \begin{cases} (v_k)_{i,j+\frac{1}{2}}^n & \text{for } \text{IFL}=2 \\ -(v_k)_{i,j+\frac{1}{2}}^n & \text{for } \text{IFL}=3 \end{cases} \end{aligned}$$

Similarly, if top cell (i, j+1) or bottom cell (i, j-1), is a rigid cell and if free-slip boundary conditions (IFL = 2) or no-slip boundary conditions (IFL = 3) are imposed, then for all n,

$$\begin{aligned}
 (u_k)_{i+\frac{1}{2},j\pm 1}^{n+1} &= \begin{cases} (u_k)_{i+\frac{1}{2},j}^n & \text{for } \text{IFL}=2 \\ -(u_k)_{i+\frac{1}{2},j}^n & \text{for } \text{IFL}=3 \end{cases} \\
 (v_k)_{i,j\pm\frac{1}{2}}^{n+1} &= 0
 \end{aligned}$$

If any one of the cells $(i, j+1)$, $(i+1, j)$, $(i-1, j)$, or $(i, j-1)$ is both a rigid cell and a corner cell, then, for all free-slip, no-slip and partial-slip boundary conditions, all its velocity components located at the center are set equal to zero.

A.8.2 Inflow Boundary Cells. For each inflow opening a pressure, P_g , velocities u_k and v_k , and volume fractions, ϵ_k for all phases ($k=g,\ell,s$), temperature of all phases T_k ($k=g,\ell,s$), composition, y_k^i ($k=g,\ell,s$), and solids granular temperature Θ_k must be specified in the input data, as necessary.

The types of inflow boundary conditions used are inflow prescribed (IFL 5); or inflow pressure prescribed, (IFL = 6). The pressure, P_g , is required for -both (IFL = 5) and (IFL = 6) to compute the mass fluxes. However, when the inflow pressure is prescribed, the radial or axial velocity components are computed using momentum equations.

A.8.3 Outflow Boundary Cells. For each outflow opening, a pressure, P_g may be specified at the beginning of each computation cycle. For constant boundary conditions, the value of P_g is that specified in the input data. The types of outflow boundary conditions used are: pressure prescribed outflow with particulate outflow (IFL=7), or no particulate outflow (IFL = 8); or continuous outflow (IFL = 4).

For both pressure-specified and continuous outflow boundary conditions, the volume fractions, for a given cell of an outflow opening, is obtained by reflection. The

tangential components of the velocities are set to zero. A numerical screen is used to keep particulate phases from leaving the fluidized bed for (IFL = 8).

Therefore, at the top outflow boundary, where $(i,j)=(I, JB2)$; $(i+1/2, j)=(I, JB2)$; $(i, j+1/2)=(I, JB2)$; $(i, j-1)=(I, JB2)$; and $(i, j-1/2)=(I, JB1)$, we have,

$$(P_g)_{i,j}^{n+1} = \begin{cases} P_0 & \text{for } IFL=7/8 \\ (P_g)_{i,j-1}^n & \text{for } IFL=4 \end{cases}$$

$$(\epsilon_k)_{i,j}^{n+1} = \begin{cases} (\epsilon_k)_{i,j-1}^n & \text{for } IFL=4/7 \\ 1 & k=g \text{ for } IFL=8 \\ 0 & k \neq g \end{cases}$$

$$(u_k)_{i+\frac{1}{2},j}^{n+1} = 0$$

$$(v_g)_{i,j+\frac{1}{2}}^{n+1} = (\epsilon_g \rho_g)_{i,j-1}^{n+1} (v_g)_{i,j-\frac{1}{2}}^{n+1} / (\epsilon_g \rho_g)_{i,j}^{n+1}$$

$$(v_k)_{i,j+\frac{1}{2}}^{n+1} = \begin{cases} (\epsilon_k \rho_k)_{i,j-1}^{n+1} (v_k)_{i,j-\frac{1}{2}}^{n+1} / (\epsilon_k \rho_k)_{i,j}^{n+1} & \text{for } IFL=4/7 \\ 0 & \text{for } IFL=8 \end{cases}$$

At the right outflow boundary, where $(i, j)=(IB2, j)$; $(i+1/2, j)=(IB2, J)$; $(i-1, j)=(IB1, J)$; $(i-1/2, j)=(IB1, J)$; and $(i-1, j)=(IB1, J)$.

$$(P_g)_{i,j}^{n+1} = \begin{cases} P_0 & \text{for } IFL=7/8 \\ (P_g)_{i-1,j}^n & \text{for } IFL=4 \end{cases}$$

$$(\epsilon_k)_{i,j}^{n+1} = \begin{cases} (\epsilon_k)_{i-1,j}^n & \text{for } IFL=4/7 \\ 1 & k=g \text{ for } IFL=8 \\ 0 & k \neq g \end{cases}$$

$$(u_g)_{i+\frac{1}{2},j}^{n+1} = (\epsilon_g \rho_g)_{i-1,j}^{n+1} (u_g)_{i-\frac{1}{2},j}^{n+1} / (\epsilon_g \rho_g)_{i,j}^{n+1}$$

$$(u_k)_{i+\frac{1}{2},j}^{n+1} = \begin{cases} (\epsilon_k \rho_k)_{i-1,j}^{n+1} (u_k)_{i-\frac{1}{2},j}^{n+1} / (\epsilon_k \rho_k)_{i,j}^{n+1} & \text{for } IFL=4/7 \\ 0 & \text{for } IFL=8 \end{cases}$$

$$(v_k)_{i,j+\frac{1}{2}}^{n+1} = 0$$

A.9 Initial Conditions

At the beginning of the simulation the distribution of all field variables is given by the initial conditions. Uniform and simple non-uniform initial conditions can be specified using the input data. Fluid blocks (FL = 1) with the field variables may be defined in similar manner as the obstacle blocks are defined. Complex non-uniform initial conditions should be programmed.

The pressure in the axial direction is initialized by the gravity head of the fluid or/and particulate phases. If (IPRES = 0), pressure profile is obtained using fluid phase only and the weight of the particulate phases is not supported by fluid phase. If (IPRES = 1), the fluid phase supports the weight of the complete bed.

The pressure distribution in all fluid cells is computed recursively from the weight of the bed as follows,

$$(P_g)_{i,j} = (P_g)_{i+1,j} - g_z \delta z_{j+\frac{1}{2}} \begin{cases} (\epsilon_g \rho_g)_{i,j+\frac{1}{2}} & \text{for } \text{IPRES}=0 \\ \sum_{k=g,1}^N (\epsilon_k \rho_k)_{i,j+\frac{1}{2}} & \text{for } \text{IPRES}=1 \end{cases}$$

where pressure in the top most cell $J = JB2$ is given by $(P_g)_{i,j+1/2} = P_0$. For density of fluid, equation of state may be used. Since this density may depend on the pressure in cell (i, j) which is yet to be calculated, a quadratic equation must be solved in a general case.

In a rigorous approach, the pressure distribution in a two-dimensional region must be computed by solving the finite differenced continuity and momentum equations with initial values of volume fractions, velocities and temperatures of all required phases.

APPENDIX B
ORGANIZATION AND USER MENU OF MFREK

B.1 Program Organization

A flow chart of the program MFREK.f with the various subroutines is given in Figure B.1. An alphabetical listing of the various subroutines and their functions are as follows:

BDRY	Sets the boundary conditions - reflects cell centered quantities.
BETAS	Calculates the reciprocal derivatives of the mass residuals with respect to pressure $\overline{\beta}_{i,j} = (\partial D / \partial P_g)_{i,j}$ for iteration procedure.
CNVERT	Calculates the initial microscopic and macroscopic densities and converts temperature to enthalpies.
COMPOS	Calculates composition for each phase.
FEFLUX	Calculates fluctuating energy fluxes of kinetic theory of granular solids.
FLIC	Sets cell flags based on input data.
GRNVIS	Calculates new granular temperature and particulate properties using kinetic theory of granular solids.
GRPROP	Initializes particulate properties using kinetic theory of granular solids.
GRTEMF	Calculates granular temperature fluxes of kinetic theory.
HEATCG	Calculates the heat flux for the gas phase, using the phase conductivities.
HEATCL	Calculates the heat flux for the solid and/or liquid phases, using the phase conductivities.
ICONV	Updates the specific energies to account for the effects of convection, viscous and pressure work, and conduction.
IGIL	Updates the specific energies to account for the effects of mass momentum and energy exchange.

Figure B.1 The Program Flow Chart

IGIL	Updates the specific energies to account for the effects of mass momentum and energy exchange.
IINV	Calculates inverse of matrix with non-zero first column, first row and diagonal column.
INDX	Calculates indices for array quantities.
ITER	Performs the iterative solution of the difference equation of mass momentum and energy equations.
KDRAGG	Calculates gas-solid drag coefficients.
KDRAGS	Calculates gas-solid drag for low or high solid concentration.
MASFG	Calculates mass fluxes for the gas phase.
MASFK	Calculates mass fluxes for the solid and/or liquid phases.
MATS	Calculates the matrix component for velocity calculations.
MULTI	Calculates particle to particle interactions.
NEWP	Calculates a new estimate of advanced time pressure from three (pressure residual) points.
PROD	Calculates the mass flux at inlet and outlet for gas phase.
PROG	Controls the program flow and output.
QESOL	Solves quadratic equation.
QFLUX	Calculates heat flux from heat boundaries (around heat exchangers).
QRXN	Calculates the heat generations due to reactions.
RHEATS	Calculates the value of interface heat transfer coefficient.
RRATE	Defines the rates of reactions and heat generations.

RXN	Calculates phase changes.
SETUP	Defines the initial values of field variables in the fluid, inflow and outflow boundary cells, using the input data.
SIEGF	Calculates fluxes of specific energy for the gas phase.
SIELF	Calculates fluxes of specific energy for the solid and/or phases.
SOLUTION	Identifies rate control step (reaction or mass transfer).
TAPERD	Reads the restart file for initial conditions.
TAPEWR	Writes to a restart file.
THRCON	Calculates the thermal conductivities for the phases.
TILDE	Calculate momenta due to convection, gravity, viscous stress, solids pressure and cohesive stress (tilde quantities).
UGMOMF	Calculates fluxes of radial momentum for the gas phase.
UGVS	Calculates stress tensor terms for the gas phase in radial direction.
ULMOMF	Calculates fluxes of radial momentum for the particulate phases.
ULVS	Calculates stress tensor terms for the particulate phases in radial direction.
VELINV	Uses Gauss-Dolittle method for symmetric matrix inversion.
VELSK	Calculates velocities on the four boundaries of the cell.
VGMOMF	Calculate fluxes of axial momentum for the gas phase.
VGVS	Calculates stress tensor term for the gas phase in axial direction.
VLMOMF	Calculate fluxes of axial momentum for the particulate phases.
VLVS	Calculates stress tensor term for the particulate phases in axial direction.
VWORKL	Calculates the viscous work for the phase specific energy equation.

B.2 Setting up a Program

First of all choose an appropriate coordinate system from rectangular and cylindrical coordinates. Use the symmetries of the problem to reduce the computational region. Properly specify the inflow and outflow coordinates which can be located in any positions.

Non-uniform initial conditions, if any conditions can be programmed by modifying the subroutine SETUP.

Specify dx_i and dy_j according to the degree of resolution required. Sometimes these dimensions are governed by the size of inflow and outflow openings. However, the ratio between two adjacent increments should be within bounds,

$$\frac{1}{\overline{K}} < \frac{dx_i}{dx_{i+1}} \quad \text{or} \quad \frac{dy_j}{dy_{j+1}} < \overline{K}$$

where $\overline{K}$ is usually a number between 1 and 2 (Noye, 1978; Cebeci and Smith, 1971). The accuracy (and occasional the stability) is sensitive to the value of $\overline{K}$ used. For satisfactory results $\overline{K} < 1.15$ should be used.

The time step is limited by Courant stability condition (Courant et al. 1952),

$$\delta t < \frac{\min(\delta r_i, \delta z_j)}{\max(\text{velocities})}$$

The solution technique is a generalization of the Implicit Continuum Eulerian (ICE) method and hence there is no sonic time step limitation. However, in some cases the time step required for proper convergence could be much less than what is given by the above expression.

B.3 Input Data. A description of the input data file is given below.

B.3.1

RESTART: restart file name (no more than 10 characters).

NAME: case identifier (no more than 80 characters).

(DR(I),I=1,IB2): radial steps.

(DZ(J),J=1,JB2): axial steps.

(NSL(M),M=1,4): cell flag of bottom, left, top and right boundary.

B.3.2

NSL(1), ITHMF(1), (IOB(M,1),M=1,4): cell flag, heat flag (=1 indicate a heat sink/source block, =0 indicate a non-heat sink/source block and skip the following line), coordinates in order of left, right, bottom and top of block for the **first** block.

(The following line is optional).

QQ(1), TOBB(1), COEK(1): constant heat flux, constant temperature and heat transfer coefficient of the heat block. One must supply QQ or TOBB and COEK.

NSL(2), ITHMF(2), (IOB(M,2),M=1,4): cell flag, heat flag (=1 indicate heat sink/source block, =0 indicate a non-heat sink/source block and skip the following line), coordinates in order of left, right, bottom and top of block for the **second** block.

(The following line is optional).

QQ(2), TOBB(2), COEK(2): constant heat flux, constant temperature and heat transfer coefficient of the heat block. One must supply QQ or TOBB and COEK.

NSL(NT), ITHMF(NT), (IOB(M,NT),M=1,4): cell flag, heat flag (=1 indicate a heat sink/source block, =0 indicate a non-heat sink/source block and skip the

following line), coordinates in order of left, right, bottom and top of block for the **NT-th** block. (The following line is optional).

QQ(NT), TOBB(NT), COEK(NT): constant heat flux, constant temperature and heat transfer coefficient of the heat block. One must supply QQ or TOBB and COEK.

B.3.3

DK(NPHASE): particle diameters of particulate phases.

PHI(NPHASE): sphericity of particles.

RL(NPHASE): density of particulate phases.

CL(NPHASE): specific heat of particulate phases.

VISS(NPHASE): empirical viscosity of particulate phases. Values of VISS must be given even kinetic theory approach is used.

(KIN(K), RLKMIN(K), K=1, NPHASE): KIN=1 use kinetic theory. Kinetic theory is used only when solid concentration is more than RLKMIN.

B.3.4

UIO(1), VIO(1), PIO(1), THIO(1), TEMIO(1): gas velocities in x-direction, y-direction, pressure, volume fraction and temperature of the **first** block.

UPIO(K,1), VPIO(K,1), THPIO(K,1), TEMPIO(K,1): k-phase velocities in x-direction, y-direction, volume fraction and temperature of the **first** block.

UIO(2), VIO(2), PIO(2), THIO(2), TEMIO(2): gas velocities in x-direction, y-direction, pressure, volume fraction and temperature of the **second** block.

UPIO(K,2), VPIO(K,2), THPIO(K,2), TEMPIO(K,2): k-phase velocities in x-direction, y-direction, volume fraction and temperature of the **second** block.

UIO(NO), VIO(NO), PIO(NO), THIO(NO), TEMIO(NO): gas velocities in x-direction, y-direction, pressure, volume fraction and temperature of the **NO-th** block.

UPIO(K,NO), VPIO(K,NO), THPIO(K,NO), TEMPIO(K,NO): k-phase velocities in x-direction, y-direction, volume fraction and temperature of the **NO-th** block.

B.3 5

IHEAT, IRXN: IHEAT=1 for non-isothermal case, IRXN=1 for case with reactions.

B.3 6. If IRXN is not 1, skip this section.

WM(JXN): molecular weights.

IHO(K), IHE(K): # of homogeneous and heterogeneous reactions for phase k (k starts from gas, solid1, solid2, ...)

HHO(K,IX), (AHO(K,IX,JX), JX=1,JXN): reaction heat, stoichiometric coefficients of IX-th homo-reaction in phase k. If IHO(K)=0, skip this line.

HHE(K,IX), (AHE(K,IX,JX), JX=1,JXN): reaction heat, stoichiometric coefficients of IX-th hetero-reaction in phase k (exclude gas). If IHE(K)=0, skip this line.

((YIO(K,JX,1),JX=1,JXN),K=1,NPHS1): initial composition for phase k (gas, solid1, solid2, ...) in the **first** block.

((YIO(K,JX,1),JX=1,JXN),K=1,NPHS1): initial composition for phase k (gas, solid1, solid2, ...) in the **second** block.

((YIO(K,JX,1),JX=1,JXN),K=1,NPHS1): initial composition for phase k (gas, solid1, solid2, ...) in the **NO-th** block.

B.3.7

ITD: =0 no restart; =2 read initial data from a restart file.

TIME, TSTOP, DT: start time, stop time, time interval for simulation.

TPR, TDUMP: time interval for outputting data, time interval for updating restart file.

GRAVY, GRAVX: gravity in x- and y- directions. For inclined bed, they depend on the inclined angle. The other external forces such as electrostatic-force may be added to these terms and modification of GRAVY(K), GRAVX(K) may be needed if these forces are differ from phase to phase.

B.4 Common Block. In addition, some of constants must be given as input data in the INCLUDE file "mfrek.com."

Line 2:

IB2=2 + # of radial cells.

JB2=2+ # of axial cells.

NPHASE=# of particulate phases.

Line 3:

NIN=# of inflow blocks.

NOUT=# of outflow blocks.

NFL=# of flow (performing calculation) blocks.

NOBS=# of non-flow blocks.

JXN=# of species. Non-reacting phase such as catalysts may be treated as a specie.

THICK=thickness of the 2-d bed, used for mass balancing and flowrate computing.

Line 4:

ITC=0 Cartesian/1 cylindrical coordinates.

WMYGAS=average molecular weight of continuous phase (gas).

THMIN=minimum volume fraction allowed for continuous phase (gas).

CRES=restitution coefficient used in kinetic theory. It is sensitive to the value of granular temperature.

Line 5:

CG=specific heat of continuous phase (gas).

VISF=viscosity of continuous phase (gas).

Line 6:

RST=left wall position (usually =0).

IPRE=0 uniform initial pressure distribution; =1 weight-depended initial pressure distribution.

Line 7:

C17= 8.314E+7 if gas as continuous phase; =0 if liquid as continuous phase.

C18=0 if gas as continuous phase; =WMYGAS/DENSITY OF LIQUID if liquid as continuous phase.

Important: if modified "mfrek.f" the common variables must be added to the COMMON BLOCKS of "mfrek.com."

B.4 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 dimensions of array variables needs to be changed when the input data is changed. The indices shown in the following table and the respective dimensions to be used are: I= (1--IB2); J = (1--JB2); IJ = (1--IB2xJB2); JX=(1--JXN) where JXN is total number of species; K = (1--N) where N is total number of particulate phases and M = (1--Nx(N+I)/2). The rest are left unchanged unless otherwise specified.

Symbol	Description
ABETA(IJ)	$1/\beta_{i,j}$
AHE(K,IX,JX)	Stoichiometric coefficient of heterogeneous reaction.
AHO(K,IX,JX)	Stoichiometric coefficient of homogeneous reaction.
AKGO	Reference fluid thermal conductivity.
AKG(IJ)	Thermal conductivity of gas phases.
AKL(K,IJ)	Thermal conductivity of particulate phases.
APP(M,IJ)	$\delta t \sum_{l=1}^N (\beta_{kl})_{i,j}$ as diagonal element.
AR(I)	$dx_{i+1}/2dx_{i+1/2}$.
ARL(K)	Receptacle of phase density.
AU(N,N)	Components of matrix A for cell (i+1/2, j).
AU1(N,N)	Components of matrix A for cell (i-1/2, j).
AV(N,N)	Components of matrix A for cell (i, j+1/2).

AV1(N,N)	Components of matrix A for cell (j, j-1/2).
AH(N,N)	Components of matrix A_h for cell (i, j).
AZ(J)	$dy_{j+1}/2dy_{j+1/2}$.
BR(I)	1-AR(I)
BU(N,N)	Components of vector B_u for cell (i+1/2,j).
BU1(N,N)	Components of vector B_u for cell (i-1/2, j).
BV(N,N)	Components of vector B_v for cell (i, j+1/2).
BV1(N,N)	Components of vector B_v for cell (j, j-1/2).
BH(N,N)	Components of vector B_h for cell(i, j)
BZ(J)	1-AZ(J)
C1--C21	
C1, C2	Reference temperature of gas, particulate phases.
C17	=8.314E+7 if gas as continuous phase; =0 if liquid as continuous phase.
C18	=0 if gas as continuous phase; =WMYGAS/DENSITY OF LIQUID if liquid as continuous phase.
CG	Specific heat of gas phase.
CL(K)	Specific heat of particulate phases.
CPHI(K)	Solids volume fraction at maximum packing.
COE(NC)	Heat conductivity of heat boundary for cell (i,j).
COEK(NT)	Heat conductivity of heat boundary for block N.
COHF(0:1000)	Cohesive stress values.

CONV(IJ)	Pressure iteration convergence criteria computed in BETAS.
CRES	Coefficient of restitution.
D1, D2, D3	Values of D_{ij} .
DG	D_{ij} - Tolerance in fluid continuity equation.
DGG(IJ)	Store D_{ij} for composition calculation.
DK(K)	d_k - Characteristic diameter of k-th phase.
DOTM(K)	Phase change.
DOTMJ(K,JX)	Species change in phase k.
DR(I)	dx_i - Variable cell size in x- (or r-) direction at i-th column.
DRP(I)	$dx_{i+1/2}$ - Variable cell size in x- (or r-) direction at i+1/2-th column.
DT	dt - Time increment.
DTOBDR(IJ)	$dt/x_{i+1/2}/dx_{i+1/2}$.
DTODR(IJ)	dt/dx_i .
DTODRP(IJ)	$dt/dx_{i+1/2}$.
DTODZ(IJ)	dt/dy_j .
DTODZP(IJ)	$dt/dy_{j+1/2}$.
DTORDR(IJ)	$dt/x_i/dx_i$.
DZ(J)	dy_j - Variable cell size in y- (or z-) direction at j-th row.
DZP(J)	$dy_{j+1/2}$ - Variable cell size in z-direction at j + 1/2-th row.
DI, D2, D3	Values of D_{ij} .
EGFB(I)	Flux of gas phase specific energy across the bottom boundary of cell (i, j).

EGFL	Flux of gas phase specific energy across left boundary of cell (i,j).
EGFR	Flux of gas phase specific energy across right boundary of cell (i,j).
EGFT	Flux of gas phase specific energy across top boundary of cell (i,j).
ELFB(K,I)	Flux of k-phase specific energy across the bottom boundary of cell (i, j).
ELFL(K)	Flux of k-phase specific energy across left boundary of cell (i,j).
ELFR(K)	Flux of k-phase specific energy across right boundary of cell (i,j).
ELFT(K)	Flux of k-phase specific energy across top boundary of cell (i,j).
EPSG	Value for convergence criteria (default = 10^{-5}).
FLUX	Gas mass flux at outlet.
GAMMA(K,IJ)	Collision energy dissipation rate.
GCDIL(K)	k_{dil} - dilute phase granular conductivity.
GCON(K,IJ)	Granular thermal conductivity for particulate phases.
GRAVX	Gravity in x-direction.
GRAVY	Gravity in y-(or z-) direction.
GTH(IK)	Solids stress values.
HFGB(I)	Conductive heat flux of gas phase across the bottom boundary of cell (i,j).
HFGL	Conductive heat flux of gas phase across left boundary of cell (i,j).
HFGR	Conductive heat flux of gas phase across right boundary of cell (i,j).
HFGT	Conductive heat flux of gas phase across top boundary of cell (i,j).
HFLB(K,I)	Conductive heat flux of k-phase across the bottom boundary of cell (i,j).

HFL(K)	Conductive heat flux of k-phase across left boundary of cell (i,j).
HFLR(K)	Conductive heat flux of k-phase across right boundary of cell (i,j).
HFLT(K)	Conductive heat flux of k-phase across top boundary of cell (i,j).
HHE	Hetero-reaction heat.
HHO	Homo-reaction heat.
I	I - Computing mesh column index (x- or r- direction).
IB	Number of cell in the radial direction excluding the two fictitious columns along the boundaries; (IB = IB2 -2).
IB1	(IB1 = IB2 -1).
IB2	Total number of cells in the radial direction.
IB1JB2	IB2xJB2 -1.
IB2JB1	IB2x(JB2 - 1) = IB2JB2 - IB2.
IB2JB2	IB2xJB2.
IFL(IJ)	Cell flags set up in FLIC.
IHE	Number of hetero-reactions.
IHEAT	=1 for solving energy equations.
IHO	Number of homo-reactions.
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).
IJBR	Index of cell centered quantities associated with cell (i+1,j-1).
IJL	Index of cell centered quantities associated with cell (i-1,j).
IJM	Index of cell (i,j-1).

IJP	Index of cell $(i,j+1)$.
IJR	Index of cell centered quantities associated with cell $i+1, j$.
IJRR	Index of cell centered quantities associated with cell $(i+2,j)$.
IJT	Index of cell centered quantities associated with cell $(i,j+1)$.
IJTL	Index of cell centered quantities associated with cell $(i-1,j+1)$.
IJTR	Index of cell centered quantities associated with cell $(i+1,j+1)$.
IJTT	Index of cell centered quantities associated with cell $(i,j+2)$.
IMJ	Index of cell $i-1,j$.
IMJM	Index of cell $i-1,j-1$.
IMJP	Index of cell $(i-1,j+1)$.
INDC()	Storage of IPJ, IMJ, IJP, IJM, IPJP, IMJP, IPJM, and IMJM for each (i,j) .
INDS()	Storage of IJR, IJL, IJT, IJB, IJTR, IJTL, IJBR, IJRR, and IJTT for each (i,j) .
IOB(M,N)	Coordinates of nth block.
IPJ	Index of cell $(i+1,j)$.
IPJM	Index of cell $(i+1,j-1)$.
IPJP	Index of cell $(i+1,j+1)$.
IPRES	=0 initial uniform pressure, =1 initial pressure distribution depended on weight of bed.
IRXN	=0 or 1 represented case without or with reactions.
ITD	=0 no restart, =2 read initial data from a restart file.

ITHMF	=0 non heat block, =1 heat sink/source block.
J	Computing mesh index (z-direction).
JB	(JB = JB2 -2).
JB1	(JB1 = JB2 -1).
JB2	Total number of cells in the axial direction.
JXN	Total number of species.
K	Index for the particulate phases.
KEY(4)	Used in COMPOS.
Kin(K)	=1 Using kinetic theory for viscosity; =0 Using empirical viscosity.
KV	Index for continuous phase.
LT	=NPHS1 Total number of phases ($\geq$ NPHASE).
MAXIT	Maximum number of iterations.
NAME	Problem identifier.
NC	Total number of cells ($\geq$ IB2JB2).
NF	$\text{NPHASE} \times (\text{NPHASE} + 1) / 2$.
NFL	Number of fluid blocks.
NPHASE	Number of particulate phases.
NH	Number of inflow openings.
NI	=IB2
NIT	Iteration counter used in ITER.
NJ	=JB2
NOBS	Number of rigid blocks.

NO	Number of inflow, outflow and fluid blocks.
NOOUT	Number of outflow exits.
NP	=NPHS1
NPHS1	Total number of phases ($\geq$ NPHASE).
NPHASE	Number of particulate phases.
NS	Total number of solids phases ($\geq$ NSOLID).
NSOLID	Number of particulate phases.
NSL(M)	Boundary flag.
NSO(N)	Block flag.
NT	Number of total blocks.
P(IJ)	Gas pressure in cell (i,j).
P1, P2, P3	Values of $(P_g)_{ij}$.
PHI(K)	sphericity of k-th phase.
PI	=3.1415927.
PIO(N)	Initial pressure of block n.
PN(IJ)	Gas pressure in cell (i,j) at time level n.
PS(K,ij)	$(P_k)_{ij}$ Solids Pressure in cell (i,j).
PVISC(K)	Input viscosity.
Q, QF, QFLOW	Heat flux from heat sink/source.
QGEN	Heat flux due to reactions.
R(I)	dx_i - Radial coordinate of the center of cell (i,j).
RB(I)	$dx_{i+1/2}$ - Radial coordinate of the right boundary of cell (i, j).

RDR(I)	$1/ dx_i$.
RDRP(I)	$1/dx_{i+1/2}$
RDZ(J)	dy_j .
RDZP(J)	$dy_{j+1/2}$.
RGFRK(K,IJ)	Flux of macroscopic density across the right boundary of cell (i, j).
RGFTK(K,IJ)	Flux of macroscopic density across the top boundary of cell (i, j).
RGFRKY(K,IJ)	Flux of macroscopic density across the right boundary of cell (i, j).
RGFTKY(K,IJ)	Flux of macroscopic density across the top boundary of cell (i, j).
RGP(IJ)	Macroscopic density of the gas for cell (i, j).
RGPN(IJ)	Macroscopic density of the gas for cell (i, j) at time level n.
RHE	Rate of hetero-reaction.
RHO	Rate of homo-reaction.
RHEAT(K,IJ)	Interfacial heat transfer function for gas and particulate phases for cell (i, j).
RKPG(K,IJ)	Gas-particulate phase drag for cell (i, j).
RL(K)	Microscopic particulate phase density.
RLFRK(K,IJ)	Flux of macroscopic density across the right boundary of cell (i, j).
RLFTK(K,IJ)	Flux of macroscopic density across the top boundary of cell (i, j).
RLFRKY(K,IJ)	Flux of macroscopic density across the right boundary of cell (i, j).
RLFTKY(K,IJ)	Flux of macroscopic density across the top boundary of cell (i, j).
RLK(K,IJ)	Phase density for cell (i, j).
RLKN(K,IJ)	Phase density for cell (i, j) at time level n.

RLX(K,IJ)	Mixture density for cell (i, j).
ROG(IJ)	Microscopic density of the gas for cell (i, j).
RRB(I)	$1/x_{i+1/2}$.
RRIDR(I)	$1/x_i/dx_i$.
RRIDRP(I)	$1/x_{i+1/2}/dx_{i+1/2}$.
RST	Radial distance of left boundary from origin.
RUG(IJ)	$\overline{\epsilon_g \rho_g u_g}$.
RVG(IJ)	$\overline{\epsilon_g \rho_g v_g}$.
RUK(K,IJ)	$\overline{\epsilon_k \rho_k u_k}$.
RVK(K,IJ)	$\overline{\epsilon_k \rho_k v_k}$.
SIEG(IJ)	$(H_g)_{ij}$ - Specific energy of the gas phases for cell (i,j).
SIEGN(IJ)	$(H_g)_{ij}^n$ - Specific energy of the gas phases at time level n.
SIEL(K,IJ)	$(H_k)_{ij}$ - Specific energy of the k-phases for cell (i,j).
SIELN(K,IJ)	$(H_k)_{ij}^n$ - Specific energy of the k-phases at time level n.
SILM(K,IJ)	$(\text{grad } v_k)_{ij}$.
SVREL(K)	Square of relative velocity of particulate phase.
SUB	Product of radius, volume fraction, and shear stress.
SUGAR	Product of vol. fraction and azimuthal stress.
SUGL	Product of radius, volume fraction, and radial stress.
SUGR	Same as SUGL, but evaluated at cell location (i+1,j).
SUGT	Same as SUGB, but evaluated at cell location (i+1/2,j+1/2).
SULB(K)	Product of radius, volume fraction, and shear stress.

SULC(K)	Product of vol. fraction and azimuthal stress.
SULL(K)	Product of radius, volume fraction, and radial stress.
SULR(K)	Same as SULL(K), but evaluated at cell location (i+1,j).
SULT(K)	Same as SULB(K), but evaluated at cell location (i+1/2,j+1/2).
SVGB	Product of volume fraction, and axial stress.
SVGL	Product of radius, volume fraction, and axial stress.
SVGR	Same as SVGL, but evaluated at cell (i+1/2, j+1/2).
SVGT	Same as SVGB, but evaluated at cell location (i, j+1).
SVLB(K)	Product of volume fraction, and axial stress.
SVLL(K)	Product of radius, volume fraction, and axial stress.
SVLR(K)	Same as SVLL(K), but evaluated at cell (i+1/2, j+1/2).
SVLT(K)	Same as SVLB(K), but evaluated at cell location (i, j+1).
TARGET	Used in the pressure iteration to provide over or under relaxation.
TDUMP	Time interval for restart file.
TEMIO(K,NT)	Initial temperature of phase k of block n.
TG(IJ)	Gas temperature.
TGN(IJ)	Gas temperature at time level n.
TH(IJ)	Gas volume fractions for cell (i,j).
THICK	Thickness of the 2-d slice (for PROD).
THIO(K,NT)	Initial volume fraction for phase k of block n.
THN(IJ)	Gas volume fractions for cell (i,j) at time level n.
THMIN	Minimum allowed gas volume fraction.

TIME	Initial time.
TL(K,IJ)	$(T_k)_{i,j}$ - Phase k temperature for cell (i,j).
TLN(K,IJ)	$(T_k)_{i,j}$ - Phase k temperature for cell (i,j) at time level n.
TPR	Time interval for output.
TOB(IJ)	Temperature of obstacle cells (i,j).
TOBB(IJ)	Temperature of obstacle block.
TSK(K,IJ)	Granular temperature.
TSKCB(K)	Granular conductive flux across the bottom boundary of cell (i,j).
TSKCL(K)	Granular conductive flux across the left boundary of cell (i,j).
TSKCR(K)	Granular conductive flux across the right boundary of cell (i,j).
TSKCT(K)	Granular conductive flux across the top boundary of cell (i,j).
TSKFB(K)	Flux of granular kinetic energy across the bottom boundary of cell (i,j).
TSKFL(K)	Flux of granular kinetic energy across left boundary of cell (i,j).
TSKFR(K)	Flux of granular kinetic energy across the right boundary of cell (i,j).
TSKFT(K)	Flux of granular kinetic energy across top boundary of cell (i,j).
TSKN(K,IJ)	Granular temperature at time level n.
TSKIO(K,NT)	Initial granular temperature.
TSTOP	Final time.
UG(IJ)	Gas radial velocity.

UGFB (IJ)	Radial momentum flux for the gas phase across the bottom boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
UGFL(IJ)	Radial momentum flux for the gas phase across the left boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
UGFR(IJ)	Radial momentum flux for the gas phase across the right boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
UGFT(IJ)	Radial momentum flux for the gas phase across the top boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
UIO(NT)	Initial radial gas velocity.
UPIO(K,NT)	Initial radial k-phase velocity.
ULFB (K,IJ)	Radial momentum flux for k-phases across the bottom boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
ULFL(K,IJ)	Radial momentum flux for k-phases across the left boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
ULFR(K,IJ)	Radial momentum flux for k-phases across the right boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.

ULFT(K,IJ)	Radial momentum flux for k-phases across the top boundary of the radial momentum control volume centered about the point $(i+1/2,j)$.
VG(IJ)	Gas axial velocity.
VGFB(K,IJ)	Axial momentum flux for gas phase across the bottom boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VGFL(K,IJ)	Axial momentum flux for gas phase across the left boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VGFR(K,IJ)	Axial momentum flux for gas phase across the right boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VGFT(K,IJ)	Axial momentum flux for gas phase across the top boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VIO(NT)	Initial gas velocity.
VISBL(K,IJ)	Particulate bulk viscosity for cell (i,j) .
VISCL(K,IJ)	Particulate shear viscosity for cell (i,j) .
VISDIL(K)	Dilute phase granular viscosity.
VISF	Gas viscosity.
VISS, VISSD	Empirical solid viscosity.
VLFB(K,IJ)	Axial momentum flux for the phases across the bottom boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.

VLFL(K,IJ)	Axial momentum flux for the phases across the left boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VLFR(K,IJ)	Axial momentum flux for the phases across the right boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VLFT(K,IJ)	Axial momentum flux for the phases across the top boundary of the radial momentum control volume centered about the point $(i,j+1/2)$.
VK(K,IJ)	Axial velocity for cell (i,j) .
VPIO(K,NT)	Initial k-phase axial velocity.
VREL(K)	Velocity of solid phase relative to gas phase.
VWLM(K,IJ)	Viscous work generated by symmetric gradient of velocity term.
VWLS(K,IJ)	Viscous work generated by correction term.
WM(JX)	Molecular weight of specie j.
WMYGAS	Average gas or liquid molecular weight (used in no reaction case).
YIO(K,JX,N)	Initial compositions of k-phase.
Y(K,JX,IJ)	Weight fraction of specie j in k-phase.
YN(K,JX,IJ)	Weight fraction of specie j in k-phase at time level n.

APPENDIX C
SOURCE FORTRAN CODE OF MFREK
AND A SAMPLE INPUT DATA

C.1 MFREK.F

```

C-----+
C      MULTIPHASE
C      FLOW SIMULATOR WITH
C      REACTIONS,
C      ENERGY EQUATIONS AND
C      KINETIC THEORY.
C
C      MODIFIED BY
C      YUANXIANG WU & DR. DIMITRI GIDASPOW
C
C      ILLINOIS INSTITUTE OF TECHNOLOGY
C      CHICAGO, ILLINOIS 60616
C-----+
PROGRAM MFREK
INCLUDE 'mfrek.com'
CHARACTER*10 RESTART
CHARACTER*80 NAME
READ(5,'(A)')RESTART
OPEN(9,FILE=RESTART,FORM='UNFORMATTED',TYPE='UNKNOWN')
READ(5,'(A)')NAME
WRITE(6,'(A)') SIMULATION - ',NAME
READ(5,*)(DR(I),I=1,IB2)
READ(5,*)(DZ(J),J=1,JB2)
WRITE(6,210)ITC,IB2,JB2,DR(1),DZ(1)
WRITE(6,215)(DR(I),I=1,IB2)
WRITE(6,216)(DZ(J),J=1,JB2)
READ(5,*)(NSL(M),M=1,4)
WRITE(6,220)(NSL(M),M=1,4)
WRITE(6,230)NIN,NOUT,NFL,NOBS
WRITE(6,240)
DO 5 N=1,NT
READ(5,*)NSO(N),ITHMF(N),(IOB(M,N),M=1,4)
WRITE(6,245)NSO(N),(IOB(M,N),M=1,4)
IF(ITHMF(N).EQ.1)THEN
READ(5,*)QQ(N),TOBB(N),COEK(N)
ENDIF
5  CONTINUE
READ(5,*)(DK(K),K=1,NPHASE)
READ(5,*)(PHI(K),K=1,NPHASE)
READ(5,*)(RL(K),K=1,NPHASE)
READ(5,*)(CL(K),K=1,NPHASE)
READ(5,*)(VISS(K),K=1,NPHASE)
READ(5,*)(KIN(K),RLKMIN(K),K=1,NPHASE)

```

```

WRITE(6,250)
WRITE(6,255)(DK(K),RL(K),CL(K),PHI(K),K=1,NPHASE)
READ(5,*)UIO(1),VIO(1),PIO(1),THIO(1),TEMIO(1)
READ(5,*)(UPIO(K,1),VPIO(K,1),THPIO(K,1),TEMPPIO(K,1)
1,TSKIO(K,1),K=1,NPHASE)
WRITE(6,260)
WRITE(6,265)UIO(1),VIO(1),PIO(1),THIO(1),TEMIO(1)
WRITE(6,270)
WRITE(6,275)(K,UPIO(K,1),VPIO(K,1),THPIO(K,1),TEMPPIO(K,1),K=1
1,NPHASE)
WRITE(6,280)
DO 10 N=2,NO
READ(5,*)UIO(N),VIO(N),PIO(N),THIO(N),TEMIO(N)
READ(5,*)(UPIO(K,N),VPIO(K,N),THPIO(K,N),TEMPPIO(K,N),
1TSKIO(K,N),K=1,NPHASE)
10 WRITE(6,285)UIO(N),VIO(N),PIO(N),THIO(N),TEMIO(N)
WRITE(6,290)
DO 15 N=2,NO
15 WRITE(6,295)(K,UPIO(K,N),VPIO(K,N),THPIO(K,N),TEMPPIO(K,N),
1K=1,NPHASE)
READ(5,*)IHEAT,IRXN
IF(IRXN.EQ.1)THEN
READ(5,*)(WM(JX),JX=1,JXN)
WRITE(6,*)
WRITE(6,*)'MOLECULAR WEIGHT'
WRITE(6,*)' CO CO2 H2 CH4 CH3OH N2'
WRITE(6,*)'WM=',(WM(JX),JX=1,JXN)
WRITE(6,*)
WRITE(6,*)'REACTION HEAT & STOICHIOMETRIC COEFF.'
DO 16 K=1,NPHASE+1
WRITE(6,*)
READ(5,*)IHO(K),IHE(K)
WRITE(6,*)'PHASE',K-1,', # OF HOMO-REACTIONS=',IHO(K),
1', # OF HETERO-REACTIONS=',IHE(K)
IF(IHO(K).NE.0)THEN
WRITE(6,*)' HOMO-REACTIONS'
DO 17 IX=1,IHO(K)
READ(5,*)HHO(K,IX),(AHO(K,IX,JX),JX=1,JXN)
17 WRITE(6,*)' REACTION',IX,' COEF.',(AHO(K,IX,JX),JX=1,JXN)
ENDIF
IF(IHE(K).NE.0)THEN
WRITE(6,*)' HETERO-REACTIONS'
DO 21 IX=1,IHE(K)
READ(5,*)HHE(K,IX),(AHE(K,IX,JX),JX=1,JXN)
21 WRITE(6,*)' REACTION',IX,' COEF.',(AHE(K,IX,JX),JX=1,JXN)

```

```

      ENDIF
16  CONTINUE
      WRITE(6,*)
      WRITE(6,*)'COMPOSITIONS'
      WRITE(6,*)
      DO 19 N=1,NO
      DO 19 K=1,NPHASE+1
      READ(5,*)(YIO(K,JX,N),JX=1,JXN)
19  WRITE(6,519)N,K-1,(YIO(K,JX,N),JX=1,JXN)
519  FORMAT(' BLOCK',I2,', PHASE ',I1,',',6G11.4)
      ENDIF
      READ(5,*)ITD
      READ(5,*)TIME,TSTOP,DT
      READ(5,*)TPR,TDUMP
      WRITE(6,300)ITD,TIME,TSTOP,DT,TPR,TDUMP
      READ(5,*)GRAVX,GRAVY
      WRITE(6,310)GRAVY,GRAVX
      WRITE(6,*)
      DO 30 J=1,JB2
      DO 30 I=1,IB2
      IJ=I+(J-1)*IB2
      TOB(IJ)=0.0
      COE(IJ)=0.0
30  QF(IJ)=0.0
      DO 22 N=1,NT
      IF(ITHMF(N).NE.1)GOTO 22
      DO 23 J=IOB(3,N),IOB(4,N)
      DO 23 I=IOB(1,N),IOB(2,N)
      IJ=I+(J-1)*IB2
      TOB(IJ)=TOBB(N)
      COE(IJ)=COEK(N)
23  QF(IJ)=QQ(N)
22  CONTINUE
      DO 27 N=1,NT
      IF(ITHMF(N).NE.1) GOTO 27
C   WRITE(6,*)'HEAT SOURCE/SINK (T-OBSTACLE/FLUX)'
      DO 24 J=1,JB2
      DO 24 I=1,IB2
      IJ=I+(J-1)*IB2
24  Q(IJ)=QF(IJ)+TOB(IJ)
C   DO 26 IJ=IB2JB2,IB2,-IB2
C26  WRITE(6,25)(Q(IL),IL=IJ-IB1,IJ)
25  FORMAT(6(1X,G12.4))
      GOTO 29
27  CONTINUE

```



```

29  CONTINUE
C
    REWIND(9)
    DO 500 K=1,NPHASE
    VISSD(K)=VISS(K)
    DO 500 IJ=1,IB2JB2
500  VISBL(K,IJ)=0.0
    TTIM=0.0
    IF (ITD.EQ.2) CALL TAPERD
    DO 20 K=1,NPHASE
20   ARL(K)=1./RL(K)
    CALL FLIC
    CALL SETUP
    CALL PROG
    STOP
210  FORMAT(/' 1. GEOMETRY/' A. COORDINATES
    1(CART=0, CYLIND=1, SPHER=2)
    1='I3/' B. MESH SIZE, IB2='I3,6X,'JB2='I3/'
    1C. CELL SIZE'
    2,', DR='1PE12.4,' DZ='1PE12.4)
215  FORMAT(7X,'DR='1/8(2X,F6.3))
216  FORMAT(7X,'DZ='1/8(2X,F6.3))
220  FORMAT(/' 2. CELL FLAGS/' A. BOUNDARIES (FREE-SLIP=2
    1NO-SLIP=3)'
    1/7X,'BOTTOM='I3,' LEFT='I3,' TOP='I3,' RIGHT='I3)
230  FORMAT(' B. INLETS ='I3,' OUTLETS ='I3,' FLUIDS ='I3,
    1' OBSTACLES ='I3)
240  FORMAT(' SLIP',10X,'-----COORDINATES-----')
245  FORMAT(5X,I3,4(4X,I5))
250  FORMAT(/' 3. PARTICULATE PHASE DATA/7X,'DIAMETER',4X,
    1'MACROSCOPIC',
    1' DENSITY',4X,' HEAT CAPACITY',4X,'SPHERICITY')
255  FORMAT((2(6X,G10.3),13X,G10.3,7X,G10.3))
260  FORMAT(/' 4. INITIAL AND BOUNDARY DATA/' A. INITIAL DATA',
    1' (FLUID)/11X,'UO',13X,'VO',13X,'PO',12X,'THO',11X,'TEMGO')
265  FORMAT((2X,5(4X,1PE11.4)))
270  FORMAT(' B. INITIAL DATA (SOLID)/8X,'PHASE',9X,
    1'UKO',11X,'VKO',10X,'THKO',9X,'TEMKO')
275  FORMAT((8X,I3,3X,4(3X,1PE11.4)))
280  FORMAT(' C. INFLOW - OUTFLOW DATA (FLUID)/10X,'UIO',12X,
    1'VIO',12X,'PIO',12X,'THIO',11X,'TEMIO')
285  FORMAT((2X,5(4X,1PE11.4)))
290  FORMAT(' D. INFLOW - OUTFLOW DATA (SOLID)/8X,'PHASE',9X,
    1'UPIO',11X,'VPIO',10X,'THPIO',9X,'TEMPIO')
295  FORMAT((8X,I3,3X,4(4X,1PE11.4)))

```

```

300  FORMAT(/' 5. CONTROL/' 3X,'A. DUMP AND RESTART,  ITD='
      1,I3/3X,
      1'B. TIME  TSTART=',1PE11.4,' TSTOP=',1PE11.4,' DT=',
      11PE11.4/
      1' C. PRINTING AND PLOTTING,  TPR=',1PE11.4,' TDUMP=',
      11PE11.4)
310  FORMAT(/' 6. GRAVITY/'  A. GRAVY=',1PE15.7,'  GRAVX='
      1,1PE15.7)
      END

```

C -----BDRY

```

      SUBROUTINE BDRY
      INCLUDE 'mfrek.com'

```

C

```

      DO 200 J=2,JB1
      DO 200 I=2,IB1
      IJ=I+(J-1)*IB2
      IF(IFL(IJ).EQ.1) THEN
      RGNP(IJ)=RGP(IJ)
      THN(IJ)=TH(IJ)
      SIEGN(IJ)=SIEG(IJ)
      TG(IJ)=C1+SIEG(IJ)/CG
      IF(IHEAT.EQ.0)TG(IJ)=TEMIO(1)
      CALL ROGY(IJ)
      AKG(IJ)=8.67D5*(TG(IJ)/1400.0)**1.786
      DO 9 K=1,NPHASE
      SIELN(K,IJ)=SIEL(K,IJ)
      IF (RLK(K,IJ).EQ.0.0) THEN
      ° TL(K,IJ)=0.0
      ELSE
      TL(K,IJ)=C2+SIEL(K,IJ)/CL(K)
      ENDIF
      IF(IHEAT.EQ.0)TL(K,IJ)=TEMPIO(K,1)
      CALL THRCON
      TSKN(K,IJ)=TSK(K,IJ)
9    RLKN(K,IJ)=RLK(K,IJ)
      IPJ=IJ+1
      IJP=IJ+IB2
      IMJ=IJ-1
      IJM=IJ-IB2
      IPJP=IJP+1
      IMJP=IJP-1
      IPJM=IJM+1
      NFLR=IFL(IPJ)
      NFLTR=IFL(IPJP)
      NFLT=IFL(IJP)

```

```

IF(NFLR.EQ.4.OR.NFLR.GE.7) THEN
  N1=IJ
  N2=IPJ
  IF(NFLR.EQ.4.OR.UG(N1).GT.0.)THEN
    RGFR(N2)=RGFR(N1)
    RGP(N2)=RGP(N1)
    RUG(N2)=RUG(N1)
    TH(N2)=TH(N1)
    TG(N2)=TG(N1)
    DO 14 JX=1,JXN
      YN(1,JX,N2)=YN(1,JX,N1)
14    Y(1,JX,N2)=Y(1,JX,N1)
      DO 15 K=1,NPHASE
        RLFRK(K,N2)=RLFRK(K,N1)
        RLK(K,N2)=RLK(K,N1)
        TSK(K,N2)=TSK(K,N1)
        TSKN(K,N2)=TSK(K,N1)
        RUK(K,N2)=RUK(K,N1)
        DO 13 JX=1,JXN
          YN(K+1,JX,N2)=YN(K+1,JX,N1)
13    Y(K+1,JX,N2)=Y(K+1,JX,N1)
15    TL(K,N2)=TL(K,N1)
      IF(NFLR.GE.7)THEN
        IJ=N2
        CALL CNVERT
        IJ=N1
      ELSE
        P(N2)=P(N1)
      ENDIF
    ENDIF
    IF(NFLTR.GE.4)THEN
      VG(N2)=VG(N1)
      DO 17 K=1,NPHASE
17    VK(K,N2)=VK(K,N1)
      ENDIF
      DO 18 K=1,NPHASE
18    UK(K,N2)=UK(K,N1)
      UG(N2)=UG(N1)
      IF (NFLR.EQ.8)THEN
        DO 20 K=1,NPHASE
          UK(K,N2)=0.0
20    VK(K,N2)=0.0
        ENDIF
      ENDIF
    IF(NFLT.EQ.4.OR.NFLT.GE.7) THEN

```

```

N1=IJ
N2=IJP
IF(NFLT.EQ.4.OR.VG(N1).GT.0.)THEN
  RGFT(N2)=RGFT(N1)
  RGP(N2)=RGP(N1)
  RVG(N2)=RVG(N1)
  TH(N2)=TH(N1)
  TG(N2)=TG(N1)
  DO 24 JX=1,JXN
    YN(1,JX,N2)=YN(1,JX,N1)
24  Y(1,JX,N2)=Y(1,JX,N1)
    DO 25 K=1,NPHASE
      RLFTK(K,N2)=RLFTK(K,N1)
      TSK(K,N2)=TSK(K,N1)
      TSKN(K,N2)=TSK(K,N1)
      RLK(K,N2)=RLK(K,N1)
      RVK(K,N2)=RVK(K,N1)
      DO 23 JX=1,JXN
        YN(K+1,JX,N2)=YN(K+1,JX,N1)
23  Y(K+1,JX,N2)=Y(K+1,JX,N1)
25  TL(K,N2)=TL(K,N1)
      IF(NFLT.GE.7)THEN
        IJ=N2
        CALL CNVERT
        IJ=N1
      ELSE
        P(N2)=P(N1)
      ENDIF
    ENDIF
    IF(NFLT.GE.4) THEN
      UG(N2)=0.0
      DO 27 K=1,NPHASE
27  UK(K,N2)=0.0
      ENDIF
      RGP(N2)=RGP(N1)
      VG(N2)=VG(N1)
C  VG(N2)=RGP(N1)*VG(N1)/RGP(N2)
      DO 28 K=1,NPHASE
28  VK(K,N2)=VK(K,N1)
      IF (NFLT.EQ.8)THEN
        DO 30 K=1,NPHASE
30  UK(K,N2)=0.0
        VK(K,N2)=0.0
      ENDIF
    ENDIF
  ENDIF

```

C OBSTACLES

```

      IF(IFL(IJ).EQ.1)THEN
      IF (NFLR .EQ. 2) THEN
      VG(IPJ)=VG(IJ)
      DO 91 K=1,NPHASE
91    VK(K,IPJ)=VK(K,IJ)
      ELSEIF (NFLR .EQ. 3 ) THEN
      VG(IPJ)=0.0
      DO 92 K=1,NPHASE
92    VK(K,IPJ)=0.0
      ENDIF
      IF (NFLT .EQ. 2) THEN
      UG(IJP)=UG(IJ)
      DO 93 K=1,NPHASE
93    UK(K,IJP)=UK(K,IJ)
      ELSEIF (NFLT .EQ. 3) THEN
      UG(IJP)=0.0
      DO 94 K=1,NPHASE
94    UK(K,IJP)=0.0
      ENDIF
      NFLL=IFL(IMJ)
      NFLTL=IFL(IMJP)
      IF (NFLL .EQ. 2) THEN
      VG(IMJ)=VG(IJ)
      DO 95 K=1,NPHASE
95    VK(K,IMJ)=VK(K,IJ)
      ELSEIF (NFLL .EQ. 3) THEN
      VG(IMJ)=0.0
      DO 96 K=1,NPHASE
96    VK(K,IMJ)=0.0
      ENDIF
      NFLB=IFL(IJM)
      NFLBR=IFL(IPJM)
      IF (NFLB .EQ. 2) THEN
      UG(IJM)=UG(IJ)
      DO 97 K=1,NPHASE
97    UK(K,IJM)=UK(K,IJ)
      ELSEIF (NFLB .EQ. 3) THEN
      UG(IJM)=0.0
      DO 98 K=1,NPHASE
98    UK(K,IJM)=0.0
      ENDIF
      ENDIF
      ENDIF
200  CONTINUE

```

RETURN
END

C -----BETAS

SUBROUTINE BETAS
INCLUDE 'mfrek.com'

C
PARAMETER (EPSG=1.0D-4)
DO 10 J=2,JB1
DO 10 I=2,IB1
IJ=I+(J-1)*IB2
IF(IFL(IJ).NE.1) GOTO 10
CALL INDXA
IF(IFL(IPJ).EQ.1.OR.IFL(IPJ).EQ.4.OR.IFL(IPJ).GE.7)THEN
RIG=RB(I)*(AR(I)*TH(IJ)+BR(I)*TH(IJR))
ELSE
RIG=0.
ENDIF
IF(IFL(IMJ).NE.2.AND.IFL(IMJ).NE.3.AND.IFL(IMJ).NE.5)THEN
EFL=RB(I-1)*(BR(I-1)*TH(IJ)+AR(I-1)*TH(IJL))
ELSE
EFL=0.
ENDIF
IF(IFL(IJP).EQ.1.OR.IFL(IJP).EQ.4.OR.IFL(IJP).GE.7)THEN
TOP=AZ(J)*TH(IJ)+BZ(J)*TH(IJT)
ELSE
TOP=0.
ENDIF
IF(IFL(IJM).EQ.1.OR.IFL(IJM).EQ.4.OR.IFL(IJM).EQ.6)THEN
BOT=BZ(J-1)*TH(IJ)+AZ(J-1)*TH(IJB)
ELSE
BOT=0.
ENDIF
CONV(IJ)=EPSG*RGP(IJ)
RBETA=TH(IJ)*ROG(IJ)/P(IJ)+DTODZ(J)*DTODZ(J)*(TOP+BOT)+
1DTODR(I)*DTODR(I)*(RIG+EFL)
ABETA(IJ)=1./RBETA
10 CONTINUE
RETURN
END

C -----CNVERT

SUBROUTINE CNVERT
INCLUDE 'mfrek.com'

C
CALL ROGY(IJ)

```

RGP(IJ)=ROG(IJ)*TH(IJ)
RGPN(IJ)=RGP(IJ)
SIEG(IJ)=(TG(IJ)-C1)*CG
SIEGN(IJ)=SIEG(IJ)
AKG(IJ)=8.67D5*(TG(IJ)/1400.0)**1.786
DO 10 K=1,NPHASE
TSKN(K,IJ)=TSK(K,IJ)
RLKN(K,IJ)=RLK(K,IJ)
SIEL(K,IJ)=(TL(K,IJ)-C2)*CL(K)
SIELN(K,IJ)=SIEL(K,IJ)
10 CONTINUE
RETURN
END

```

C -----COMPOS

```

SUBROUTINE COMPOS
C SOLVE THE COMPOSITION FOR EACH PHASE
INCLUDE 'mfrek.com'
IF(IRXN.EQ.0)RETURN
DO 2 J=2,JB1
DO 2 I=2,IB1
IJ=I+(J-1)*IB2
IF(IFL(IJ).NE.1)GOTO 2
CALL INDXA
DO 10 K=1,LT
K1=K-1
IF(K.EQ.1)GOTO 6
IF(RLK(K1,IJ).GT.0.0)GOTO 6
DO 5 JX=1,JXN
5 Y(K,JX,IJ)=0.0
GOTO 10
6 CALL RRATE
IF(K.EQ.1)THEN
C CALL MASFGY
IF(UG(IMJ).GE.0.)THEN
RGFRY(IMJ)=UG(IMJ)*RGP(IJL)*RB(I-1)
KEY(2)=IJL
ELSE
RGFRY(IMJ)=UG(IMJ)*RGP(IJ)*RB(I-1)
KEY(2)=IJ
ENDIF
IF(VG(IJM).GE.0.)THEN
RGFTY(IJM)=VG(IJM)*RGP(IJB)
KEY(4)=IJB
ELSE

```

```

    RGFTY(IJM)=VG(IJM)*RGP(IJ)
    KEY(4)=IJ
    ENDIF
C   ENTRY MASFGAY
    IF(UG(IJ).GE.0.)THEN
    RGFRY(IJ)=UG(IJ)*RGP(IJ)*RB(I)
    KEY(1)=IJ
    ELSE
    RGFRY(IJ)=UG(IJ)*RGP(IJR)*RB(I)
    KEY(1)=IJR
    ENDIF
    IF(VG(IJ).GE.0.)THEN
    RGFTY(IJ)=VG(IJ)*RGP(IJ)
    KEY(3)=IJ
    ELSE
    RGFTY(IJ)=VG(IJ)*RGP(IJT)
    KEY(3)=IJT
    ENDIF
    ELSE
C   CALL MASFKY
    IF(UK(K1,IMJ).GE.0.)THEN
    RLFRKY(K1,IMJ)=UK(K1,IMJ)*RLK(K1,IJL)*RB(I-1)
    KEY(2)=IJL
    ELSE
    RLFRKY(K1,IMJ)=UK(K1,IMJ)*RLK(K1,IJ)*RB(I-1)
    KEY(2)=IJ
    ENDIF
    IF(VK(K1,IJM).GE.0.)THEN
    RLFTKY(K1,IJM)=VK(K1,IJM)*RLK(K1,IJB)
    KEY(4)=IJB
    ELSE
    RLFTKY(K1,IJM)=VK(K1,IJM)*RLK(K1,IJ)
    KEY(4)=IJ
    ENDIF
C
C   ENTRY MASFKAY
    IF(UK(K1,IJ).GE.0.)THEN
    RLFRKY(K1,IJ)=UK(K1,IJ)*RLK(K1,IJ)
    KEY(1)=IJ
    ELSE
    RLFRKY(K1,IJ)=UK(K1,IJ)*RLK(K1,IJR)
    KEY(1)=IJR
    ENDIF
    IF(VK(K1,IJ).GE.0.)THEN
    RLFTKY(K1,IJ)=VK(K1,IJ)*RLK(K1,IJ)

```



```

KEY(3)=IJ
ELSE
RLFTKY(K1,IJ)=VK(K1,IJ)*RLK(K1,IJT)
KEY(3)=IJT
ENDIF
ENDIF
THY=0.0
DO 25 JX=1,JXN
DOTMJ(K,JX)=0.0
DO 30 IX=1,IHO(K)
30 DOTMJ(K,JX)=DOTMJ(K,JX)+RHO(K,IX)*AHO(K,IX,JX)
DO 27 IX=1,IHE(K)
27 DOTMJ(K,JX)=DOTMJ(K,JX)+RHE(K2,IX)*AHE(K2,IX,JX)
IF(K.EQ.1)THEN
SUMN=WM(JX)*DOTMJ(1,JX)*DT+RGPN(IJ)*YN(1,JX,IJ)
1+DGG(IJ)*YN(1,JX,IJ)
SUMD=TH(IJ)*ROG(IJ)
IF(KEY(1).EQ.IJ)THEN
SUMD=SUMD+DTORDR(I)*RGFRY(IJ)
ELSE
SUMN=SUMN-DTORDR(I)*RGFRY(IJ)*YN(1,JX,IJR)
ENDIF
IF(KEY(2).EQ.IJ)THEN
SUMD=SUMD-DTORDR(I)*RGFRY(IMJ)
ELSE
SUMN=SUMN+DTORDR(I)*RGFRY(IMJ)*YN(1,JX,IJL)
ENDIF
IF(KEY(3).EQ.IJ)THEN
SUMD=SUMD+DTODZ(J)*RGFTY(IJ)
ELSE
SUMN=SUMN-DTODZ(J)*RGFTY(IJ)*YN(1,JX,IJT)
ENDIF
IF(KEY(4).EQ.IJ)THEN
SUMD=SUMD-DTODZ(J)*RGFTY(IJM)
ELSE
SUMN=SUMN+DTODZ(J)*RGFTY(IJM)*YN(1,JX,IJB)
ENDIF
Y(1,JX,IJ)=SUMN/SUMD
ELSE
SUMN=WM(JX)*DOTMJ(K,JX)*DT+RLKN(K1,IJ)*YN(K,JX,IJ)
SUMD=RLK(K1,IJ)
IF(KEY(1).EQ.IJ)THEN
SUMD=SUMD+DTORDR(I)*RLFRKY(K1,IJ)
ELSE
SUMN=SUMN-DTORDR(I)*RLFRKY(K1,IJ)*YN(K,JX,IJR)

```

```

ENDIF
IF(KEY(2).EQ.IJ)THEN
SUMD=SUMD-DTORDR(I)*RLFRKY(K1,IMJ)
ELSE
SUMN=SUMN+DTORDR(I)*RLFRKY(K1,IMJ)*YN(K,JX,IJL)
ENDIF
IF(KEY(3).EQ.IJ)THEN
SUMD=SUMD+DTODZ(J)*RLFTKY(K1,IJ)
ELSE
SUMN=SUMN-DTODZ(J)*RLFTKY(K1,IJ)*YN(K,JX,IJT)
ENDIF
IF(KEY(4).EQ.IJ)THEN
SUMD=SUMD-DTODZ(J)*RLFTKY(K1,IJM)
ELSE
SUMN=SUMN+DTODZ(J)*RLFTKY(K1,IJM)*YN(K,JX,IJB)
ENDIF
Y(K,JX,IJ)=SUMN/SUMD
ENDIF

C
IF(Y(K,JX,IJ).LE.1D-10)Y(K,JX,IJ)=0.0
THY=THY+Y(K,JX,IJ)
25  CONTINUE
IF(ABS(THY-1.0).LE.1D-10)GOTO 10
C NORMALIZE THE COMPOSITION,
C PUT THE ERROR TO A MAJOR COMPONENT SUCH AS #6
IF(K.EQ.1)THEN
Y(K,6,IJ)=1.0-(THY-Y(K,6,IJ))
ELSE
Y(K,6,IJ)=1.0-(THY-Y(K,6,IJ))
ENDIF
10  CONTINUE
2   CONTINUE
3   DO 9 IJ=1,IB2JB2
IF(IFL(IJ).NE.1)GOTO 9
DO 8 K=1,LT
DO 8 JX=1,JXN
YN(K,JX,IJ)=Y(K,JX,IJ)
8   CONTINUE
9   CONTINUE
RETURN
END

C -----FEFLUX
SUBROUTINE FEFLUX
INCLUDE 'mfrek.com'

```

```

C
C CALCULATE FLUCTUATING ENERGY FLUXES OF KINETIC THEORY
C
  TSKCB(K)=(TSKN(K,IJ)-TSKN(K,IJB))
  1*(BZ(J-1)*GCON(K,IJ)+AZ(J-1)*GCON(K,IJB))*RDZP(J-1)
  TSKFB(K,I)=1.5*TSKFB(K,I)-TSKCB(K)
C
  ENTRY FEFLUXA
  IF(IFL(IMJ).NE.1)THEN
    TSKCL(K)=(TSKN(K,IJ)-TSKN(K,IJL))*RB(I-1)
    1*(BR(I-1)*GCON(K,IJ)+AR(I-1)*GCON(K,IJL))*RDRP(I-1)
    TSKFL(K)=1.5*TSKFL(K)-TSKCL(K)
  ENDIF
C
  ENTRY FEFLUXB
  TSKCR(K)=(TSKN(K,IJR)-TSKN(K,IJ))*RB(I)
  1*(BR(I)*GCON(K,IJR)+AR(I)*GCON(K,IJ))*RDRP(I)
  TSKFR(K)=1.5*TSKFR(K)-TSKCR(K)
C
  TSKCT(K)=(TSKN(K,IJT)-TSKN(K,IJ))
  1*(BZ(J)*GCON(K,IJT)+AZ(J)*GCON(K,IJ))*RDZP(J)
  TSKFT(K)=1.5*TSKFT(K)-TSKCT(K)
  RETURN
  END
C -----FLIC
  SUBROUTINE FLIC
  INCLUDE 'mfrek.com'
  IJ=0
  DO 150 J=1,JB2
  DO 150 I=1,IB2
  IJ=IJ+1
  IFL(IJ)=1
  IF (J.EQ.JB2) THEN
    IFL(IJ)=NSL(3)
  ELSEIF (J.EQ.1) THEN
    IFL(IJ)=NSL(1)
  ENDIF
  IF (I.EQ.1) THEN
    IFL(IJ)=NSL(2)
  ELSEIF (I.EQ.IB2) THEN
    IFL(IJ)=NSL(4)
  ENDIF
150  CONTINUE
  DO 300 N=1,NT
  DO 300 I=IOB(1,N),IOB(2,N)

```

```

DO 300 J=IOB(3,N),IOB(4,N)
IJ=I+(J-1)*IB2
IFL(IJ)=NSO(N)
300 CONTINUE
IF(IFL(IB1JB2).EQ.4.AND.IFL(IB2JB1).EQ.4)IFL(IB2JB2)=4
IF(IFL(IB1JB2).EQ.7.AND.IFL(IB2JB1).EQ.7)IFL(IB2JB2)=7
RETURN
END

```

C -----GRNVIS

```

SUBROUTINE GRNVIS
INCLUDE 'mfrek.com'

```

C

```

C CALCULATE NEW GRANULAR TEMPERATURES
C AND SOLIDS PROPERTIES
C USING KINETIC THEORY OF GRANULAR SOLIDS
C

```

```

CALL VWORKL
DO 10 J=2,JB1
DO 10 I=2,IB1
IJ=I+(J-1)*IB2
IF(IFL(IJ).EQ.1)THEN
IMJ=INDS(IJ,2)
IJM=INDS(IJ,4)
DO 5 K=1,NPHASE
IF(KIN(K).NE.1)GOTO 5

```

```

C GRANULAR KINETIC THEORY IS APPLIED
C ONLY IF (EF-EF,MIN) > 1.E-10
CWW IF(RLK(K,IJ).GT.RLKMIN(K).AND.
CWW 1(TH(IJ)-THMIN).GT.1.E-10)THEN
IF(RLK(K,IJ).GT.RLKMIN(K))THEN

```

C

```

THL(K,IJ)=ARL(K)*RLK(K,IJ)
TSKS=TSK(K,IJ)**0.5

```

C CALCULATE RADIAL DISTRIBUTION FUNCTION & OTHER PARAMETERS

```

G0=1.0/(1.0-(THL(K,IJ)/(1.0-THMIN))**(1./3.))

```

```

CS1=(1.0+CRES)*THL(K,IJ)*G0
CS2=3.0*(1.0-CRES)*RLK(K,IJ)*CS1

```

C

```

CS3=RLK(K,IJ)/(RLK(K,IJ)+DMFP(K))
CS4=2.*VISDIL(K)/((1.+CRES)*G0)
RSQRTPI=1.0/SQRT(PI)
GAMMA(K,IJ)=CS2*(4.0*RSQRTPI/DK(K)*TSKS-SILM(K,IJ))
PS(K,IJ)=RLK(K,IJ)*(1.0+2.0*CS1)

```

C

```

VISCD=CS4*(1.+8*CS1)**2

```

```

VISBL(K,IJ)=(4./3.)*RLK(K,IJ)*DK(K)*CS1/PI**0.5
VISCL(K,IJ)=VISCD+0.6*VISBL(K,IJ)
C
C GRANULAR CONDUCTIVITY
GCON(K,IJ)=3.75*CS4*(1.+1.2*CS1)**2*TSKS
1+1.5*VISBL(K,IJ)*TSKS
ELSE
GCON(K,IJ)=0.0
ENDIF
5 CONTINUE
ENDIF
10 CONTINUE
C
C FINAL SOLUTION OF GRANULAR TEMPERATURE
DO 100 J=2,JB1
DO 100 I=2,IB1
IJ=I+(J-1)*IB2
IF(IFL(IJ).EQ.1)THEN
CALL INDX
C
DO 20 K=1,NPHASE
IF(KIN(K).NE.1)GOTO 20
CALL GRTEMF
CW IF(RLK(K,IJ).GT.RLKMIN(K).AND.
CW 1(TH(IJ)-THMIN).GT.1.E-10)THEN
IF(RLK(K,IJ).GT.RLKMIN(K))THEN
AP0=-(1.5*RLK(K,IJ)*TSK(K,IJ)-DTODZ(J)*(TSKFT(K)
1-TSKFB(K,I)-DTORDR(I)*(TSKFR(K)-TSKFL(K)))
AP1=-(VISCL(K,IJ)*VWLS(K,IJ)+(VISBL(K,IJ)
1-(2./3.)*VISCL(K,IJ))*VWLM(K,IJ))
CWW D.G. INDICATES THE DRAG IN THE KINETIC EQS
C MUST BE ELIMINATED
AP2=(1.5*RLK(K,IJ)+DT*(3.0*RKPG(K,IJ)*0.00001
1+PS(K,IJ)*SILM(K,IJ)+GAMMA(K,IJ)))
C
CALL QESOL(AP0,AP1,AP2,TSKS)
IF(TSKS.LT.0.0)TSKS=0.0
IF(TSKS.GT.1000.0)TSKS=1000.0
TSK(K,IJ)=TSKS*TSKS
C
C SOLID PHASE BULK AND SHEAR VISCOSITIES
VISBL(K,IJ)=VISBL(K,IJ)*TSKS
VISCL(K,IJ)=VISCL(K,IJ)*TSKS
C
C SOLIDS PHASE PRESSURE

```

```

    PS(K,IJ)=PS(K,IJ)*TSK(K,IJ)
    ELSE
    TSK(K,IJ)=0.0
    VISBL(K,IJ)=0.0
    VISCL(K,IJ)=0.0
    PS(K,IJ)=0.0
    ENDIF
    TSKFL(K)=TSKFR(K)
    TSKFB(K,I)=TSKFT(K)
20    CONTINUE
C
    ENDIF
100    CONTINUE
    RETURN
    END

C -----GRPROP
    SUBROUTINE GRPROP
    INCLUDE 'mfrek.com'
C
C INITIALIZE SOLIDS PROPERTIES USING GRANULAR KINETIC THEORY
C
C GRANULAR KINETIC THEORY IS APPLIED
C ONLY IF (EF-EF,MIN) > 1.E-10
CWW IF(RLK(K,IJ).GT.RLKMIN(K).AND.
CWW 1(TH(IJ)-THMIN).GT.1.E-10)THEN
    IF(RLK(K,IJ).GT.RLKMIN(K))THEN
    THL(K,IJ)=ARL(K)*RLK(K,IJ)
    TSKS=TSK(K,IJ)**0.5
    G0=1.0/(1.0-(THL(K,IJ)/(1.0-THMIN))**(1./3.))
    CS1=(1.0+CRES)*THL(K,IJ)*G0
    CS2=3.0*(1.0-CRES)*RLK(K,IJ)*CS1
C    CS3=RLK(K,IJ)/(RLK(K,IJ)+DMFP(K))
    CS4=2.*VISDIL(K)/((1.+CRES)*G0)
    VISCD=CS4*(1.+8*CS1)**2
    VISBL(K,IJ)=(4./3.)*RLK(K,IJ)*DK(K)*CS1*TSKS/PI**0.5
    VISCL(K,IJ)=VISCD*TSKS+0.6*VISBL(K,IJ)
    PS(K,IJ)=RLK(K,IJ)*(1.0+2.0*CS1)*TSK(K,IJ)
    GCON(K,IJ)=3.75*CS4*(1.+1.2*CS1)**2*TSKS
    1+1.5*VISBL(K,IJ)
    ELSE
    VISBL(K,IJ)=0.0
    VISCL(K,IJ)=0.0
    PS(K,IJ)=0.0
    GCON(K,IJ)=0.0

```

```

ENDIF
RETURN
END

```

C -----GRTEMPF

```

SUBROUTINE GRTEMP
INCLUDE 'mfrek.com'

```

C
C CALCULATES GRANULAR TEMPERATURE FLUXES OF KINETIC THEORY
C

```

IF(UK(K,IJ).GE.0.)THEN
TSKFR(K)=RLFRK(K,IJ)*TSKN(K,IJ)
ELSE
TSKFR(K)=RLFRK(K,IJ)*TSKN(K,IJR)
ENDIF
IF(VK(K,IJ).GE.0.)THEN
TSKFT(K)=RLFTK(K,IJ)*TSKN(K,IJ)
ELSE
TSKFT(K)=RLFTK(K,IJ)*TSKN(K,IJT)
ENDIF
IF(IFL(IMJ).NE.1)GOTO 1
IF(IFL(IJM).NE.1)GOTO 2
CALL FEFLUXB
RETURN

```

C
1

```

IF(UK(K,IMJ).GE.0.)THEN
TSKFL(K)=RLFRK(K,IMJ)*TSKN(K,IJL)
ELSE
TSKFL(K)=RLFRK(K,IMJ)*TSKN(K,IJ)
ENDIF
IF(IFL(IJM).NE.1)GOTO 2
CALL FEFLUXA
RETURN

```

C
2

```

IF(VK(K,IJM).GE.0.)THEN
TSKFB(K,I)=RLFTK(K,IJM)*TSKN(K,IJB)
ELSE
TSKFB(K,I)=RLFTK(K,IJM)*TSKN(K,IJ)
ENDIF
CALL FEFLUX
RETURN
END

```

C -----HEATCG

```

SUBROUTINE HEATCG

```

INCLUDE 'mfrek.com'

C

C CALCULATES THE HEAT FLUXES FROM GAS CONDUCTIVITY

C

```

HFGB=(BZ(J-1)*AKG(IJ)+AZ(J-1)*AKG(IJB))*
1*(BZ(J-1)*TH(IJ)+AZ(J-1)*TH(IJB))*
1*(TG(IJ)-TG(IJB))*RDZP(J-1)
ENTRY HEATGA
HFGL=RB(I-1)*(BR(I-1)*AKG(IJ)+AR(I-1)*AKG(IJL))
1*(BR(I-1)*TH(IJ)+AR(I-1)*TH(IJL))*
1*(TG(IJ)-TG(IJL))*RDRP(I-1)
ENTRY HEATGB
HFGR=RB(I)*(AR(I)*AKG(IJ)+BR(I)*AKG(IJR))
1*(AR(I)*TH(IJ)+BR(I)*TH(IJR))*
1*(TG(IJR)-TG(IJ))*RDRP(I)
HFGT=(AZ(J)*AKG(IJ)+BZ(J)*AKG(IJT))
1*(AZ(J)*TH(IJ)+BZ(J)*TH(IJT))*
1*(TG(IJT)-TG(IJ))*RDZP(J)
RETURN
END

```

C

-----HEATCL

SUBROUTINE HEATCL
INCLUDE 'mfrek.com'

C

C CALCULATES THE HEAT FLUXES FROM SOLIDS CONDUCTIVITY

C

```

HFLB(K,I)=(BZ(J-1)*AKL(K,IJ)+AZ(J-1)*AKL(K,IJB))
1*(BZ(J-1)*RLK(K,IJ)+AZ(J-1)*RLK(K,IJB))*
1*(TL(K,IJ)-TL(K,IJB))*ARL(K)*RDZP(J-1)
ENTRY HEATLA
HFLK(K)=RB(I-1)*(BR(I-1)*AKL(K,IJ)+AR(I-1)*AKL(K,IJL))
1*(BR(I-1)*RLK(K,IJ)+AR(I-1)*RLK(K,IJL))
1*(TL(K,IJ)-TL(K,IJL))*ARL(K)*RDRP(I-1)
ENTRY HEATLB
HFLR(K)=RB(I)*(AR(I)*AKL(K,IJ)+BR(I)*AKL(K,IJR))
1*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1*(TL(K,IJR)-TL(K,IJ))*ARL(K)*RDRP(I)
HFLT(K)=(AZ(J)*AKL(K,IJ)+BZ(J)*AKL(K,IJT))
1*(AZ(J)*RLK(K,IJ)+BZ(J)*RLK(K,IJT))
1*(TL(K,IJT)-TL(K,IJ))*ARL(K)*RDZP(J)
RETURN
END

```

C

-----ICONV

SUBROUTINE ICONV


```

      INCLUDE 'mfrek.com'
      DIMENSION AI(NP,NP),BI(NP)
C
C  FINAL UPDATING OF THE SPECIFIC ENERGIES ( ENTHALPIES )
C
      IF(HEAT.EQ.0)RETURN
      DO 100 J=2,JB1
      DO 100 I=2,IB1
      IJ=I+(J-1)*IB2
      IF(IFL(IJ).EQ.1) THEN
      CALL INDXA
      CALL SIEGF
      CALL QRXN
      CALL QFLUX
      AI(1,1)=RGP(IJ)
      BI(1)=RGP(IJ)*SIEG(IJ)-SIEGN(IJ)*(RGP(IJ)-RGPN(IJ))-
      1DTODZ(J)*(EGFT-EGFB(I))-DTORDR(I)*(EGFR-EGFL)
C  FOR INTERNAL ENERGY (MODEL B) <<
CWU  THE WORK DONE BY THE FLUID HAS CONSIDERABLE EFFECT
C    ON ENERGY EQUATION. HERE REDUCE IT TO ITS 10%.
      1-0.1*P(IJ)*(DTODZ(J)*(VG(IJ)-VG(IJM))
      1+DTORDR(I)*(RB(I)*UG(IJ)-RB(I-1)*UG(IMJ)))
      BI(1)=BI(1)+DT*(QGEN(1)+QFLOW(1))
C >>
C  FOR ENTHALPY (MODEL B) <<
C    1+((P(IJ)-PN(IJ))+0.5*DTODZ(J)*(VG(IJ)+VG(IJM))
C    1*((AZ(J)-BZ(J-1))*P(IJ)+BZ(J)*P(IJT)-AZ(J-1)*P(IJB))
C    1+0.5*DTORDR(I)*(UG(IJ)+UG(IMJ))*(RB(I)
C    1*(BR(I)*P(IJR)+AR(I)*P(IJ))
C    1-RB(I-1)*(BR(I-1)*P(IJ)+AR(I-1)*P(IJL))))
C >>
      DO 10 K=1,NPHASE
      CALL SIELF
      KP=K+1
      RHT=-0.5*DT*RHEAT(K,IJ)
      AI(1,KP)=RHT/CL(K)
      AI(KP,1)=RHT/CG
      AI(1,1)=AI(1,1)-AI(KP,1)
      AI(KP,KP)=RLK(K,IJ)-AI(1,KP)
      RIT=(SIEG(IJ)/CG-SIEL(K,IJ)/CL(K)+TL(K,IJ)-TG(IJ))*RHT
      BI(1)=BI(1)-RIT
      BI(KP)=RLK(K,IJ)*SIEL(K,IJ)-SIELN(K,IJ)*(RLK(K,IJ)-
      1RLKN(K,IJ))+
      1RIT-DTODZ(J)*(ELFT(K)-ELFB(K,I))-DTORDR(I)*(ELFR(K)
      1-ELFL(K))

```

```

      BI(KP)=BI(KP)+DT*(QGEN(KP)+QFLOW(KP))
10    CONTINUE
      CALL IINV(NPHS1,AI,BI)
      SIEG(IJ)=BI(1)
      DO 20 K=1,NPHASE
        SIEL(K,IJ)=BI(K+1)
        ELFL(K)=ELFR(K)
20    ELFB(K,I)=ELFT(K)
        EGFB(I)=EGFT
        EGFL=EGFR
      ENDIF
100   CONTINUE
      RETURN
      END

```

C -----IGIL

```

      SUBROUTINE IGIL
      INCLUDE 'mfrek.com'
      DIMENSION AI(NP,NP),BI(NP)
      IF(IHEAT.EQ.0)RETURN
      CALL QRXN
      CALL QFLUX
      AI(1,1)=RGP(IJ)
      BI(1)=RGP(IJ)*SIEGN(IJ)
      BI(1)=BI(1)+DT*(QGEN(1)+QFLOW(1))
      DO 10 K=1,NPHASE
        KP=K+1
        RHT=-0.5*DT*RHEAT(K,IJ)
        AI(1,KP)=RHT/CL(K)
        AI(KP,1)=RHT/CG
        AI(1,1)=AI(1,1)-AI(KP,1)
        AI(KP,KP)=RLK(K,IJ)-AI(1,KP)
        RIT=(SIEGN(IJ)/CG-SIELN(K,IJ)/CL(K)+TLN(K,IJ)-TGN(IJ))*RHT
        BI(1)=BI(1)-RIT
        BI(KP)=RLK(K,IJ)*SIELN(K,IJ)+RIT
10    BI(KP)=BI(KP)+DT*(QGEN(KP)+QFLOW(KP))
      CALL IINV(NPHS1,AI,BI)
      SIEG(IJ)=BI(1)
      DO 100 K=1,NPHASE
100   SIEL(K,IJ)=BI(K+1)
      RETURN
      END

```

C -----IINV

```

      SUBROUTINE IINV(NP,A,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NP,NP),B(NP)

```

C INVERSE OF MATRIX WITH NON-ZERO FIRST COLUMN,
C FIRST ROW & DIAGONAL COLUMN

```

      DO 10 K=NP,2,-1
      IF (ABS(A(K,K)) .LE. 1.D-6) THEN
      A(1,K)=0.0
      A(K,1)=0.0
      B(K)=0.0
      ELSE
      DIV=1./A(K,K)
      A(K,1)=A(K,1)*DIV
      B(K)=B(K)*DIV
      B(1)=B(1)-A(1,K)*B(K)
      A(1,1)=A(1,1)-A(1,K)*A(K,1)
      ENDIF
10    CONTINUE
      B(1)=B(1)/A(1,1)
      DO 20 K=2, NP
20    B(K)=B(K)-A(K,1)*B(1)
      RETURN
      END

```

C -----INDX

```

      SUBROUTINE INDX
      INCLUDE 'mfrek.com'

```

```

C
      IMJP=IJ-1+IB2
      IJTL=IMJP
      IF((IFL(IMJP).EQ.2).OR.(IFL(IMJP).EQ.3)) IJTL=IJ+IB2
      IPJM=IJ+1-IB2
      IJBR=IPJM
      IF((IFL(IPJM).EQ.2).OR.(IFL(IPJM).EQ.3)) IJBR=IJ+1
      IJRR=IJ+2
      IF (I.EQ.IB1) IJRR=IJ+1
      IF((IFL(IJRR).EQ.2).OR.(IFL(IJRR).EQ.3)) IJRR=IJ+1
      IJTT=IJ+IB2+IB2
      IF (J.EQ.IB1) IJTT=IJ+IB2
      IF((IFL(IJTT).EQ.2).OR.(IFL(IJTT).EQ.3)) IJTT=IJ+IB2
      ENTRY INDXA
      IPJ=IJ+1
      IJP=IJ+IB2
      IMJ=IJ-1
      IJM=IJ-IB2
      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

```

C -----ITER

```

SUBROUTINE ITER
INCLUDE 'mfrek.com'

```

C

```

LOGICAL MUSTIT
PARAMETER (LMAX=5,OMEGA=1.05)
DO 5 IJ=1,IB2JB2
PN(IJ)=P(IJ)
TGN(IJ)=C1+SIEGN(IJ)/CG
DO 5 K=1,NPHASE
IF (RLK(K,IJ).EQ.0.0) THEN
TLN(K,IJ)=0.0
ELSE
TLN(K,IJ)=C2+SIELN(K,IJ)/CL(K)
ENDIF

```

5

```

CONTINUE
MUSTIT=.FALSE.
DO 200 NIT=1,200
DO 100 J=2,JB1
DO 100 I=2,IB1
IJ=I+(J-1)*IB2
IF (IFL(IJ).NE.1) GOTO 100
LOOP=0
KROS=-1
CALL INDXA
CALL RXN
DG=RGP(IJ)-RGPN(IJ)+DTORDR(I)*(RGFR(IJ)-RGFR(IMJ))
1+DTODZ(J)*
1(RGFT(IJ)-RGFT(IMJ))
DG=DG-DT*DOTM(1)
ADG=ABS(DG)
DGORIG=ADG
IF(ADG.LE.CONV(IJ)) GOTO 78
MUSTIT=.FALSE.
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
      DSN=SIGN(1.,D3)
      DP=-D3*ABETA(IJ)
      IF(-DP*DSN.GT.0.25*P3) DP=-0.25*DSN*P3
      P(IJ)=P(IJ)+DP
      GOTO54
13    P(IJ)=(D1*P2-D2*P1)/(D1-D2)
      ABETA(IJ)=(P1-P2)/(D1-D2)
      KROS=3
54    P3=P(IJ)
55    CONTINUE
      CALL IGIL
      TG(IJ)=C1+SIEG(IJ)/CG
      IF(IHEAT.EQ.0)TG(IJ)=TEMIO(1)
      CALL ROGY(IJ)
      DO 83 K=1,NPHASE
      IF (RLK(K,IJ).EQ.0.0) THEN
      TL(K,IJ)=0.0
      ELSE
      TL(K,IJ)=C2+SIEL(K,IJ)/CL(K)
      ENDIF
      IF(IHEAT.EQ.0)TL(K,IJ)=TEMPIO(K,1)
83    CONTINUE
      RGP(IJ)=TH(IJ)*ROG(IJ)
      CALL MATS
      CALL VELSK
      CALL MASFK
      CALL RXN
78    THX=0.0
      DO 79 K=1,NPHASE
      RLK(K,IJ)=RLKN(K,IJ)-DTORDR(I)*(RLFRK(K,IJ)-RLFRK(K,IMJ))
      1-DTODZ(J)*(RLFTK(K,IJ)-RLFTK(K,IJM))
      RLK(K,IJ)=RLK(K,IJ)+DT*DOTM(K+1)
      IF(RLK(K,IJ).LT.1.D-27)RLK(K,IJ)=0.0
79    THX=THX+RLK(K,IJ)*ARL(K)
      TH(IJ)=1.-THX
      RGP(IJ)=ROG(IJ)*TH(IJ)
      IF(ADG.LE.CONV(IJ))GOTO99

```

```

CALL MASFG
CALL RXN
DG=RGP(IJ)-RGPN(IJ)+DTORDR(I)*(RGFR(IJ)-RGFR(IMJ))+DTODZ(J)*
1(RGFT(IJ)-RGFT(IMJ))
DG=DG-DT*DOTM(1)
ADG=ABS(DG)
IF((ADG.LE.CONV(IJ)).AND.(ADG.LT.DGORIG)) GOTO 99
IF((NIT.EQ.1).AND.(LOOP.EQ.0)) DGORIG=ADG
D3=DG
LOOP=LOOP+1
IF((KROS.LT.2).AND.(LOOP.EQ.LMAX)) ABETA(IJ)=.5
1*LMAX*ABETA(IJ)
IF(LOOP.EQ.LMAX) GOTO 99
IF(KROS.EQ.3)CALL NEWP
GOTO10
99  DGG(IJ)=DG
100 CONTINUE
IF(MUSTIT) RETURN
MUSTIT=.TRUE.
IF (NIT.EQ.200) WRITE(6,*)'MAX ITERATION AT TIME = ',TIME
200 CONTINUE
RETURN
END

```

C -----KDRAGG

```

SUBROUTINE KDRAGG(DRAG,VREL)
INCLUDE 'mfrek.com'

C
DRCOE=0.44
REYN=TH(IJ)*ROG(IJ)*DK(K)*VREL*PHI(K)/VISF
IF(REYN.LT.0.001)REYN=0.001
IF(REYN.LE.1000.0) DRCOE=(24.0/REYN)*(1.0+0.15*REYN**0.687)
DRDVTH=DRCOE*VREL/TH(IJ)**2.65
IF(DRDVTH.LE.1.0D30) THEN
DRAG=0.75*RLK(K,IJ)*DRDVTH*ROG(IJ)*ARL(K)/(DK(K)*PHI(K))
ELSE
DRAG=1.0D30
ENDIF
RETURN
END

```

C -----KDRAGS

```

SUBROUTINE KDRAGS
INCLUDE 'mfrek.com'

C

```

```

DO 130 K=1,NPHASE
DV=(VG(IJ)-VK(K,IJ)+VG(IJM)-VK(K,IJM))*0.5
DU=(UG(IJ)-UK(K,IJ)+UG(IJM)-UK(K,IJM))*0.5
VREL=(DU*DU+DV*DV)**0.5
IF(TH(IJ).GE.0.8) THEN
CALL KDRAGG(DRAG,VREL)
ELSE
DENOM=DK(K)*PHI(K)*TH(IJ)
DRAG=(150.0*(1.0-TH(IJ))*VISF/DENOM+1.75*ROG(IJ)*VREL)
1*RLK(K,IJ)/(RL(K)*DENOM)
ENDIF
RKPG(K,IJ)=DRAG*RL(K)/(RL(K)-ROG(IJ))
130 CONTINUE
RETURN
END

```

C -----MASFG

```

SUBROUTINE MASFG
INCLUDE 'mfrek.com'
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

```

C -----MASFK

```

SUBROUTINE MASFK
INCLUDE 'mfrek.com'

C
DO 10 K=1,NPHASE
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,NPHASE
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

C -----MATS
SUBROUTINE MATS
INCLUDE 'mfrek.com'
CALL RXN
BU1(1)=RUG(IMJ)-DTODRP(I-1)*(P(IJ)-P(IJL))
BV1(1)=RVG(IJM)-DTODZP(J-1)*(P(IJ)-P(IJB))
AU1(1,1)=AR(I-1)*(APP(1,IJL)+RGP(IJL))
1+BR(I-1)*(APP(1,IJ)+RGP(IJ))
AV1(1,1)=AZ(J-1)*(APP(1,IJB)+RGP(IJB))
1+BZ(J-1)*(APP(1,IJ)+RGP(IJ))
AU1(1,1)=AU1(1,1)-DT*DOTM(1)
AV1(1,1)=AV1(1,1)-DT*DOTM(1)
DO 130 K=2,NPHS1
KM1=K-1

```



```

    BU1(K)=RUK(KM1,IMJ)
    BV1(K)=RVK(KM1,IJM)
    DO 110 KK=1,KM1
    KS=K*KM1/2+KK
    AU1(K,KK)=AR(I-1)*APP(KS,IJL)+BR(I-1)*APP(KS,IJ)
    AU1(KK,K)=AU1(K,KK)
    AV1(K,KK)=AZ(J-1)*APP(KS,IJB)+BZ(J-1)*APP(KS,IJ)
    AV1(KK,K)=AV1(K,KK)
110  CONTINUE
    KS=K*(K+1)/2
    AU1(K,K)=AR(I-1)*(APP(KS,IJL)+RLK(KM1,IJL))
    1+BR(I-1)*(APP(KS,IJ)+RLK(KM1,IJ))
    AV1(K,K)=AZ(J-1)*(APP(KS,IJB)+RLK(KM1,IJB))
    1+BZ(J-1)*(APP(KS,IJ)+RLK(KM1,IJ))
    AU1(K,K)=AU1(K,K)-DT*DOTM(K)
    AV1(K,K)=AV1(K,K)-DT*DOTM(K)
130  CONTINUE
    ENTRY MATSA
    CALL RXN
    BU(1)=RUG(IJ)-DTODRP(I)*(P(IJR)-P(IJ))
    BV(1)=RVG(IJ)-DTODZP(J)*(P(IJT)-P(IJ))
    AU(1,1)=AR(I)*(APP(1,IJ)+RGP(IJ))+BR(I)*(APP(1,IJR)+RGP(IJR))
    AV(1,1)=AZ(J)*(APP(1,IJ)+RGP(IJ))+BZ(J)*(APP(1,IJT)+RGP(IJT))
    AU(1,1)=AU(1,1)-DT*DOTM(1)
    AV(1,1)=AV(1,1)-DT*DOTM(1)
    DO 230 K=2,NPHS1
    KM1=K-1
    BU(K)=RUK(KM1,IJ)
    BV(K)=RVK(KM1,IJ)
    DO 210 KK=1,KM1
    KS=K*KM1/2+KK
    AU(K,KK)=AR(I)*APP(KS,IJ)+BR(I)*APP(KS,IJR)
    AU(KK,K)=AU(K,KK)
    AV(K,KK)=AZ(J)*APP(KS,IJ)+BZ(J)*APP(KS,IJT)
    AV(KK,K)=AV(K,KK)
210  CONTINUE
    KS=K*(K+1)/2
    AU(K,K)=AR(I)*(APP(KS,IJ)+RLK(KM1,IJ))
    1+BR(I)*(APP(KS,IJR)+RLK(KM1,IJR))
    AV(K,K)=AZ(J)*(APP(KS,IJ)+RLK(KM1,IJ))
    1+BZ(J)*(APP(KS,IJT)+RLK(KM1,IJT))
    AU(K,K)=AU(K,K)-DT*DOTM(K)
    AV(K,K)=AV(K,K)-DT*DOTM(K)
230  CONTINUE
C

```

```

RETURN
END

```

C -----MULTI

```

SUBROUTINE MULTI
INCLUDE 'mfrek.com'
KS=1
DO 100 K=1,NPHASE
C  CALCULATE PARTICLE TO PARTICLE INTERACTION
  KS=KS+1
  APP(KS,IJ)=-RKPG(K,IJ)*DT
  DO 98 KK=1,K-1
    KS=KS+1
    DV=(VK(K,IJ)-VK(KK,IJ)+VK(K,IMJ)-VK(KK,IMJ))*0.5
    DU=(UK(K,IJ)-UK(KK,IJ)+UK(K,IMJ)-UK(KK,IMJ))*0.5
    VREL=(DU*DU+DV*DV)**0.5
98  APP(KS,IJ)=-RLK(K,IJ)*RLK(KK,IJ)*DKF(K,KK)*VREL
100  KS=KS+1
    DO 105 K=1,NPHS1
      SUM=0.0
      DO 101 KK=1,K-1
        KS=K*(K-1)/2+KK
101  SUM=SUM+APP(KS,IJ)
      DO 102 KK=K+1,NPHS1
        KS=KK*(KK-1)/2+K
102  SUM=SUM+APP(KS,IJ)
      KS=K*(K+1)/2
105  APP(KS,IJ)=-SUM
    RETURN
  END

```

C -----NEWP

```

SUBROUTINE NEWP
INCLUDE 'mfrek.com'
C
  IF(D1.NE.D3) THEN
    PA=(D1*P3-D3*P1)/(D1-D3)
  ELSE
    PA=0.5*(P2+P3)
  ENDIF
  IF(D2.NE.D3) THEN
    PB=(D2*P3-D3*P2)/(D2-D3)
  ELSE
    PB=0.5*(P1+P3)
  ENDIF
  IF(D1*D3.GT.0.) THEN
    IF(PA.LT.P2.OR.PA.GT.P3)PA=0.5*(P2+P3)

```

```

ELSE
IF(PB.LT.P3.OR.PB.GT.P1)PB=0.5*(P1+P3)
ENDIF
P(IJ)=0.5*(PA+PB)
RETURN
END

```

C -----PROD

```

SUBROUTINE PROD
INCLUDE 'mfrek.com'
TMASSG=0.0
TMASSS=0.0
DO 10 J=1,JB2
KI=0
DO 9 JX=1,JXN
9  AYM(JX,J)=0.0
  AYMOL(JX,J)=0.0
  FLUX(J)=0.0

  DO 20 I=2,IB2-1
  IJ=I+(J-1)*IB2
  IF(IFL(IJ).EQ.3)GOTO 20

  CALL ROGY(IJ)
  WMO=0.0
  DO 7 JX=1,JXN
7  WMO=WMO+Y(1,JX,IJ)/WM(JX)
  DO 8 JX1=1,JXN
  AYM(JX1,J)=AYM(JX1,J)+TH(IJ)*ROG(IJ)*VG(IJ)
  1*Y(1,JX1,IJ)*DR(I)*THICK
  YMOL(JX1,IJ)=Y(1,JX1,IJ)/WM(JX1)/WMO
8  AYMOL(JX1,J)=AYMOL(JX1,J)+YMOL(JX1,IJ)
  KI=KI+1
  FLUX(J)=FLUX(J)+TH(IJ)*ROG(IJ)*VG(IJ)*DR(I)*THICK
  IF(J.EQ.1.OR.J.EQ.JB2)GOTO 20
  TMASSG=TMASSG+TH(IJ)*ROG(IJ)*DR(I)*DZ(J)*THICK
  TMASSS=TMASSS+RLK(1,IJ)*DR(I)*DZ(J)*THICK
20 CONTINUE
  DO 11 JX=1,JXN
  AYM(JX,J)=AYM(JX,J)/KI
11 AYMOL(JX,J)=AYMOL(JX,J)/KI
  WMO=0.0
  DO 12 JX=1,JXN
12 WMO=WMO+AYM(JX,J)/WM(JX)
  DO 13 JX=1,JXN
  IF(WMO.EQ.0.0)THEN

```

```

      AYMOL1(JX,J)=AYMOL(JX,J)
      ELSE
      AYMOL1(JX,J)=AYM(JX,J)/WM(JX)/WMO
      ENDIF
13  CONTINUE
10  CONTINUE

      AMIX(1)=TMASSG
      AMIX(2)=TMASSS
      AMIX(3)=FLUX(1)
      AMIX(4)=FLUX(JB2)
      DO 15 JX=1,JXN
15  AMIX(4+JX)=AYMOL(JX,JB2)
      DO 16 JX=1,JXN
16  AMIX(10+JX)=AYMOL1(JX,JB2)
      AMIX(17)=TIME
      RETURN
      END
C -----PROG
      SUBROUTINE PROG
      INCLUDE 'mfrek.com'
C
      TDUMP1=TIME
      TPRI=TIME
C SPECIAL APPLICATION
      VJET=4.012
C END
1  CONTINUE
      CALL BDRY
C SPECIAL APPLICATION
      CALL PROD
C END
      TPD1=TIME+0.1*DT
      IF(TPD1.GE.TPRI) THEN
      TPRI=TPRI+TPR
      IF(TIME.GE.0.0)THEN
      WRITE(6,547)TIME
      DO 600 K=1,NPHASE
      IF(KIN(K).EQ.1)THEN
      WRITE(6,*)
      WRITE(6,*)'VISCOSITY',K
      WRITE(6,*)
      DO 323 IJ=IB2JB2,IB2,-IB2
323 WRITE(6,550)(VISCL(1,IL),IL=IJ-IB1,IJ)
      WRITE(6,*)

```

```

WRITE(6,*)'P-SOLID',K
WRITE(6,*)
DO 322 IJ=IB2JB2,IB2,-IB2
322  WRITE(6,550)(PS(1,IL),IL=IJ-IB1,IJ)
WRITE(6,*)
WRITE(6,*)'GRANULAR TEMPERATURE',K
WRITE(6,*)
DO 324 IJ=IB2JB2,IB2,-IB2
324  WRITE(6,550)(TSK(1,IL),IL=IJ-IB1,IJ)
ENDIF
600  CONTINUE
WRITE(6,548)
DO 325 IJ=IB2JB2,IB2,-IB2
325  WRITE(6,550)(P(IL),IL=IJ-IB1,IJ)
DO 328 K=1,NPHASE
WRITE(6,549)K
DO 328 IJ=IB2JB2,IB2,-IB2
328  WRITE(6,550)(RLK(K,IL)*ARL(K),IL=IJ-IB1,IJ)
WRITE(6,551)
DO 331 IJ=IB2JB2,IB2,-IB2
331  WRITE(6,550)(TH(IL),IL=IJ-IB1,IJ)
WRITE(6,552)
DO 332 IJ=IB2JB2,IB2,-IB2
332  WRITE(6,550)(VG(IL),IL=IJ-IB1,IJ)
WRITE(6,553)
DO 333 IJ=IB2JB2,IB2,-IB2
333  WRITE(6,550)(UG(IL),IL=IJ-IB1,IJ)
DO 337 K=1,NPHASE
WRITE(6,556)K
DO 336 IJ=IB2JB2,IB2,-IB2
336  WRITE(6,550)(VK(K,IL),IL=IJ-IB1,IJ)
WRITE(6,557)K
DO 337 IJ=IB2JB2,IB2,-IB2
337  WRITE(6,550)(UK(K,IL),IL=IJ-IB1,IJ)
IF(IHEAT.EQ.1)THEN
WRITE(6,554)
DO 338 IJ=IB2JB2,IB2,-IB2
338  WRITE(6,550)(TG(IL),IL=IJ-IB1,IJ)
DO 339 K=1,NPHASE
WRITE(6,555)K
DO 339 IJ=IB2JB2,IB2,-IB2
339  WRITE(6,550)(TL(K,IL),IL=IJ-IB1,IJ)
ENDIF
C SPECIAL APPLICATION
C OUTPUT COMPOSITIONS (AIR PRODUCTS CASE) (OPTIONAL)

```

```

C    WRITE(6,*)'CO'
C    DO 335 IJ=IB2JB2,IB2,-IB2
C335 WRITE(6,550)(YMOL(1,IL),IL=IJ-IB1,IJ)
C    WRITE(6,*)'CO2'
C    DO 400 IJ=IB2JB2,IB2,-IB2
C400 WRITE(6,550)(YMOL(2,IL),IL=IJ-IB1,IJ)
C    WRITE(6,*)'H2'
C    DO 401 IJ=IB2JB2,IB2,-IB2
C401 WRITE(6,550)(YMOL(3,IL),IL=IJ-IB1,IJ)
C    WRITE(6,*)'N2'
C    DO 402 IJ=IB2JB2,IB2,-IB2
C402 WRITE(6,550)(YMOL(6,IL),IL=IJ-IB1,IJ)
C    WRITE(6,*)'CH3OH'
C    DO 403 IJ=IB2JB2,IB2,-IB2
C403 WRITE(6,550)(YMOL(5,IL),IL=IJ-IB1,IJ)
C    WRITE(6,*)'CH3OH IN LIQUID PHASE (KMOLE/M3)'
C    DO 404 IJ=IB2JB2,IB2,-IB2
C404 WRITE(6,550)(Y(3,5,IL)*1000.0*.70025/WM(5),IL=IJ-IB1,IJ)
C    WRITE(6,*)'H2O IN LIQUID PHASE'
C    DO 405 IJ=IB2JB2,IB2,-IB2
C405 WRITE(6,550)(Y(3,6,IL),IL=IJ-IB1,IJ)
C END
    ENDIF
    ENDIF
    IF(TPDT.GT.TSTOP.OR.TPDT.GT.TDUMP1) THEN
    CALL TAPEWR
    REWIND(9)
C SPECIAL APPLICATION
C GRADULLY INCREASE THE JET VELOCITIES
    IF(ITD.NE.2)THEN
    VJET=VJET+VJET
    IF(VJET.GE.80.24)VJET=80.24
    VG(3)=VJET
    VG(8)=VJET
    VG(13)=VJET
    ENDIF
    WRITE(6,100)AMIX(17),(AMIX(J),J=1,4),AMIX(17)
100  FORMAT(6(G11.5))
C END
    TDUMP1=TDUMP1+TDUMP
    ENDIF
    IF(TPDT.LT.TSTOP) THEN
    TIME=TIME+DT
    CALL TILDE
    CALL BETAS

```

```

      CALL ITER
      DO 11 KNN=1,NPHASE
      IF(KIN(KNN).EQ.1)CALL GRNVIS
11  CONTINUE
      CALL ICONV
      IF(IRXN.EQ.1)CALL COMPOS
      GOTO 1
      ENDIF
      RETURN
547  FORMAT(1X,/,1X,'@ TIME = ',G10.3)
548  FORMAT(1X,/,1X,'P',/)
549  FORMAT(1X,/,1X,'TH',I1,/)
550  FORMAT(6(1X,G11.4))
551  FORMAT(1X,/,1X,'THETA',/)
552  FORMAT(1X,/,1X,'VG',/)
553  FORMAT(1X,/,1X,'UG',/)
554  FORMAT(1X,/,1X,'TG',/)
555  FORMAT(1X,/,1X,'T',I1,/)
556  FORMAT(1X,/,1X,'V',I1,/)
557  FORMAT(1X,/,1X,'U',I1,/)
      END

```

C -----QESOL

```

      SUBROUTINE QESOL(AP0,AP1,AP2,XSOL)
      IMPLICIT REAL*8(A-H,O-Z)

```

```

C
C SOLVE QUADRATIC EQUATION

```

```

C
C SCALING OF COEFFICIENTS

```

```

      AAP0=ABS(AP0)
      AAP1=ABS(AP1)
      AAP2=ABS(AP2)
      APMAX=AAP2
      IF(APMAX.LT.AAP1)APMAX=AAP1
      IF(APMAX.LT.AAP0)APMAX=AAP0
      AP0=AP0/APMAX
      AP1=AP1/APMAX
      AP2=AP2/APMAX

```

```

C
      ENTRY QESOL1(AP0,AP1,AP2,XSOL)
      IF(AP0.EQ.0.0)THEN
      IF(AP2.EQ.0.0)THEN
      XSOL=0.0
      ELSE
      XSOL=-AP1/AP2

```

```

IF(XSOL.LT.0.0)XSOL=0.0
ENDIF
ELSE
IF(AP1.EQ.0.0)THEN
IF(AP2.EQ.0.0)THEN
XSOL=0.0
ELSE
DISC=-AP0/AP2
IF(DISC.LE.0.0)THEN
XSOL=0.0
ELSE
XSOL=DISC**0.5
ENDIF
ENDIF
ELSE
IF(AP2.EQ.0.0)THEN
XSOL=-AP0/AP1
IF(XSOL.LT.0.0)XSOL=0.0
ELSE
GOTO 10
ENDIF
ENDIF
ENDIF
RETURN

```

```

C
ENTRY QESOL2(AP0,AP1,AP2,XSOL)
10 CONTINUE
SAP1=AP1*AP1
DISC=SAP1-4.0*AP0*AP2
IF(DISC.LT.-1.E-4*SAP1)THEN
XSOL=0.0
ELSE
IF(DISC.LT.0.0)DISC=0.0
IF(AP1.LT.0.0)THEN
XSOL=(-AP1+DISC**0.5)/(2.0*AP2)
ELSE
XSOL=-2.0*AP0/(AP1+DISC**0.5)
ENDIF
ENDIF
RETURN
END

```

```

C -----QFLUX
SUBROUTINE QFLUX
INCLUDE 'mfrek.com'

```



```

DO 5 K=1,LT
5  QFLOW(K)=0.0
  IF(IHEAT.EQ.0)RETURN
  QFL=QF(IJM)+QF(IMJ)+QF(IPJ)+QF(IJP)
  IF(TOB(IJM).NE.0.0)QFLOW(1)=(QFL-(COE(IJM)*(TG(IJ)-TOB(IJM))
    1/DZ(J)))*TH(IJ)
  IF(TOB(IMJ).NE.0.0)QFLOW(1)=(QFL-(COE(IMJ)*(TG(IJ)-TOB(IMJ))
    1/DR(I)))*TH(IJ)
  IF(TOB(IPJ).NE.0.0)QFLOW(1)=(QFL-(COE(IPJ)*(TG(IJ)-TOB(IPJ))
    1/DR(I)))*TH(IJ)
  IF(TOB(IJP).NE.0.0)QFLOW(1)=(QFL-(COE(IJP)*(TG(IJ)-TOB(IJP))
    1/DZ(J)))*TH(IJ)
  DO 10 K=2,LT
  IF(TOB(IJM).NE.0.0)QFLOW(K)=(QFL-(COE(IJM)*(TL(K-1,IJ)-TOB(IJM))
    1/DZ(J)))*RLK(K-1,IJ)*ARL(K-1)
  IF(TOB(IMJ).NE.0.0)QFLOW(K)=(QFL-(COE(IMJ)*(TL(K-1,IJ)-TOB(IMJ))
    1/DR(I)))*RLK(K-1,IJ)*ARL(K-1)
  IF(TOB(IPJ).NE.0.0)QFLOW(K)=(QFL-(COE(IPJ)*(TL(K-1,IJ)-TOB(IPJ))
    1/DR(I)))*RLK(K-1,IJ)*ARL(K-1)
  IF(TOB(IJP).NE.0.0)QFLOW(K)=(QFL-(COE(IJP)*(TL(K-1,IJ)-TOB(IJP))
    1/DZ(J)))*RLK(K-1,IJ)*ARL(K-1)
10  CONTINUE
20  RETURN
    END

C -----QRXN
  SUBROUTINE QRXN
C  CALCULATE THE HEAT GENERATION DUE TO REACTIONS
  INCLUDE 'mfrek.com'
  IF(IRXN.EQ.0)RETURN
  CALL RRATE
  DO 10 K=1,LT
  QGEN(K)=0.0
  DO 15 IX=1,IHO(K)
15  QGEN(K)=QGEN(K)+RHO(K,IX)*HHO(K,IX)
  IF(K.NE.1)THEN
  DO 20 IX=1,IHE(K)
20  QGEN(K)=QGEN(K)+RHE(K,IX)*HHE(K,IX)
  ENDIF
10  CONTINUE
  RETURN
  END

C -----RHEATS
  SUBROUTINE RHEATS
  INCLUDE 'mfrek.com'

```

```

C
C   INTERPASE HEAT TRANSFER COEFFICIENT
C
DO 10 K=1,NPHASE
  SP=RLK(K,IJ)*(6./DK(K))*ARL(K)
  DV=(VG(IJ)-VK(K,IJ)+VG(IJM)-VK(K,IJM))*0.5
  DU=(UG(IJ)-UK(K,IJ)+UG(IMJ)-UK(K,IMJ))*0.5
  VREL=SQRT(DU*DU+DV*DV)
  REYN=DK(K)*VREL*ROG(IJ)/VISF
  PR=CG*VISF/AKG(IJ)
C   GUNN MODEL
  CRPR=PR**(1./3.)
  IF (TH(IJ) .GE. 0.8) THEN
    IF(REYN.LE.200.)THEN
      PNU=(2.+1.1*REYN**.6*CRPR)*SP
    ELSEIF(REYN.LE.2000.)THEN
      PNU=.123*(4.*REYN/DK(K))**.83*SP**.17
    ELSE
      PNU=.61*REYN**.67*SP
    ENDIF
  ELSE
    IF(REYN.LE.200.)THEN
      PNU=(2.+0.16*REYN**0.67)*SP
    ELSEIF(REYN.LE.1000.)THEN
      PNU=8.2*REYN**0.6*SP
    ELSE
      PNU=1.06*REYN**0.457*SP
    ENDIF
  ENDIF
  RHEAT(K,IJ)=PNU*AKG(IJ)/DK(K)
10  CONTINUE
  RETURN
  END
C -----ROGY
  SUBROUTINE ROGY(MIJ)
C   CALCULATE GAS DENSITY
  INCLUDE 'mfrek.com'
  IF(IRXN.EQ.1)THEN
    WMY=0.0
    DO 10 JX=1,JXN
10   WMY=WMY+Y(1,JX,MIJ)/WM(JX)
    WMY=1.0/WMY
  ELSE
    WMY=WMYGAS
  ENDIF

```

```

      ROG(MIJ)=P(MIJ)*WMY/(C17*TG(MIJ)+C18*P(MIJ))
      RETURN
      END

```

C

C

C

-----RRATE

```

      SUBROUTINE RRATE

```

```

C  DEFINE THE RATE OF REACTIONS AND HEAT OF REACTIONS

```

```

      INCLUDE 'mfrek.com'
      IF(IRXN.EQ.0)RETURN
      DO 20 MK=1,LT
      DO 10 IX=1,IHO(MK)
10    RHO(MK,IX)=0.0
      DO 20 IX=1,IHE(MK)
20    RHE(MK,IX)=0.0
      TMO=0.0
      DO 30 JX=1,JXN
30    TMO=TMO+YN(1,JX,IJ)/WM(JX)
      PATM=P(IJ)*1D-6
      PH2=PATM*YN(1,3,IJ)/WM(3)/TMO
      PCO=PATM*YN(1,1,IJ)/WM(1)/TMO
      PME=PATM*YN(1,5,IJ)/WM(5)/TMO
      T=TG(IJ)
      IF(T.LE.430.)T=430.0
      IF(T.GE.630.)T=630.0
      IF(PH2.EQ.0.0.OR.PCO.EQ.0.0)THEN
      RHO(1,1)=0.0
      ELSE
      RK0=8.93D-8*RLK(1,IJ)

```

```

C  TE CHANG'S KEQ MODEL CORRECTED BY THE DATA FROM DME REPORT

```

```

C    RKEQ=1420.8*EXP(21.225+9143.6/T-7.492*LOG(T)
C    1+4.076D-3 *T-7.161D-8*T*T)
      RKEQ=6.249/PATM**2
      RHO(1,1)=RK0*PCO**(1.0/3.0)*PH2**(2.0/3.0)
      1*(1-PME/PCO/PH2**2.0/RKEQ)
      DTO=RGPN(IJ)*TMO
      CCO=RGPN(IJ)*YN(1,1,IJ)/WM(1)
      CH2=RGPN(IJ)*YN(1,3,IJ)/WM(3)
      CME=RGPN(IJ)*YN(1,5,IJ)/WM(5)
C    CALL SOLUTION(RKEQ,DTO,CCO,CH2,CME,RHOX)
C    RHOX=RHOX/DT
      IF(RHO(1,1)*DT.GE.MIN(CCO,2*CH2))
      1RHO(1,1)=0.1*MIN(CCO,2*CH2)/DT

```

```

C   RHO(1,1)=1.98D7*EXP(-RT1)*(PH2**0.4)*(PCO**0.13)
C   1-2.15D10*EXP(-RT2)*(PME**0.13)
C   RHO(1,1)=RHO(1,1)*RLK(1,IJ)/3.6D6
C   CLM=YN(3,5,IJ)*0.70025/WM(5)
C   HME=1.49D6*EXP(-17235.0/8.314/TG(IJ))
C   CLGM=10.0*PME/HME
C   RHE(3,1)=1.2*(CLGM-CLM)*RLK(2,IJ)/0.70025
C   IF(RHE(3,1).LE.0.0)RHE(3,1)=0.0
C   IF(RHO(1,1).LE.0.0)RHO(1,1)=0.0

```

```

ENDIF

```

```

HHO(1,1)=-8.314D8*T*T*(-9143.6/T/T-7.492/T
1+4.076D-3-7.161D-8*T*2)

```

```

RETURN

```

```

END

```

```

C -----RXN

```

```

SUBROUTINE RXN

```

```

C CALCULATE PHASE CHANGES

```

```

INCLUDE 'mfrek.com'

```

```

IF(IRXN.EQ.0)RETURN

```

```

CALL RRATE

```

```

DOTM(1)=0.0

```

```

DO 10 K=2,LT

```

```

DOTM(K)=0.0

```

```

DO 20 JX=1,JXN

```

```

DO 20 IX=1,IHE(K)

```

```

20 DOTM(K)=DOTM(K)+RHE(K,IX)*WM(JX)*AHE(K,IX,JX)

```

```

DOTM(1)=DOTM(1)-DOTM(K)

```

```

10 CONTINUE

```

```

RETURN

```

```

END

```

```

C -----SETUP

```

```

SUBROUTINE SETUP

```

```

INCLUDE 'mfrek.com'

```

```

C

```

```

WRITE(6,660)

```

```

DO 1 I=1,IB1

```

```

DRP(I)=0.5*(DR(I)+DR(I+1))

```

```

RDR(I)=1.0/DR(I)

```

```

RDRP(I)=2.0/(DR(I)+DR(I+1))

```

```

AR(I)=0.5*DR(I+1)*RDRP(I)

```

```

1 BR(I)=1.0-AR(I)

```

```

DRP(IB2)=DR(IB2)

```

```

RDR(IB2)=1.0/DR(IB2)

```

```

RDRP(IB2)=RDR(IB2)

```

```

AR(IB2)=0.5
BR(IB2)=0.5
DO 2 J=1,JB1
DZP(J)=0.5*(DZ(J)+DZ(J+1))
RDZ(J)=1.0/DZ(J)
RDZP(J)=2.0/(DZ(J)+DZ(J+1))
AZ(J)=0.5*DZ(J+1)*RDZP(J)
2  BZ(J)=1.0-AZ(J)
DZP(JB2)=DZ(JB2)
RDZ(JB2)=1.0/DZ(JB2)
RDZP(JB2)=RDZ(JB2)
AZ(JB2)=0.5
BZ(JB2)=0.5
IF(ITC.EQ.0) GO TO 5
RTC=RST-0.5*DR(1)
RTB=RST
R(1)=RTC**ITC
RB(1)=RTB**ITC
IF((RTC.LE.0.).AND.(ITC.EQ.2))R(1)=-R(1)
IF (RB(1).LT.1.D-8) THEN
RRB(1)=0.0
ELSE
RRB(1)=1.0/RB(1)
ENDIF
DO 3 I=2,IB2
RTC=RTB+0.5*DR(I)
RTB=RTB+DR(I)
R(I)=RTC**ITC
RB(I)=RTB**ITC
RRB(I)=1./RB(I)
3  CONTINUE
GOTO 10
5  CONTINUE
DO 8 I=1,IB2
R(I)=1.
RB(I)=1.
RRB(I)=1.
8  CONTINUE
10 CONTINUE
DO 11 I=1,IB2
RRIDR(I)=RDR(I)/R(I)
RRIDRP(I)=RRB(I)*RDRP(I)
DTODR(I)=DT*RDR(I)
DTODRP(I)=DT*RDRP(I)
DTORDR(I)=DT*RRIDR(I)

```

```

      DTOBDR(I)=DT*RRIDRP(I)
11  CONTINUE
      DO 12 J=1,JB2
      DTODZ(J)=DT*RDZ(J)
12  DTODZP(J)=DT*RDZP(J)
      DO 15 IJ2=IB2JB2,IB2,-IB2
      IJ1=IJ2-IB1
15  WRITE(6,650)(IFL(IKPR),IKPR=IJ1,IJ2)
C
      WRITE(6,660)
C  INITIALIZE SOLID PRESSURE AND COHESIVE STRENGTH
      DO 20 I=0,1000
      THX=I/1000.
      COHF(I)=10.**(-10.6*THX+5.5)
C  COHF(I)=0.0
C20  GTH(I)=10.**(-8.76*THX+8.49)
20  GTH(I)=10.**(-10.466*THX+8.577)
C  INITIALIZE 'INDS'
      DO 30 J=2,JB1
      DO 30 I=2,IB1
      IJ=I+(J-1)*IB2
      IPJ=IJ+1
      IJR=IPJ
      IJP=IJ+IB2
      IJT=IJP
      IMJ=IJ-1
      IJL=IMJ
      IJM=IJ-IB2
      IJB=IJM
      IJPJ=IJP+1
      IJTR=IPJP
      IF((IFL(IPJ).EQ.2).OR.(IFL(IPJ).EQ.3)) IJR=IJ
      IF((IFL(IMJ).EQ.2).OR.(IFL(IMJ).EQ.3)) IJL=IJ
      IF((IFL(IJP).EQ.2).OR.(IFL(IJP).EQ.3)) IJT=IJ
      IF((IFL(IJM).EQ.2).OR.(IFL(IJM).EQ.3)) IJB=IJ
      IF(IJ.EQ.(IB2JB1-1))IJTR=IJ
      GOTO (29,21,21,29,29,29,29),IFL(IPJP)
21  GOTO (22,24,24,22,22,22,22),IFL(IJP)
22  GOTO (25,26,26,25,25,25,25),IFL(IPJ)
24  GOTO (28,25,25,28,28,28,28),IFL(IPJ)
25  IJTR=IJ
      GOTO 29
26  IJTR=IJP
      GOTO 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,NPHASE
    DO 40 KK=1,K-1
40  DKF(K,KK)=FAC*DT*(DK(K)+DK(KK))**2/(RL(K)*
    1DK(K)**3+RL(KK)*DK(KK)**3)
    IF(ITD.NE.2)THEN
    DO 60 N=1,NO
    DO 60 J=IOB(3,N),IOB(4,N)
    JJ=(J-1)*IB2
    DO 60 I=IOB(1,N),IOB(2,N)
    IJ=I+JJ
    IF((IFL(IJ).EQ.2).OR.(IFL(IJ).EQ.3))GOTO 60
    P(IJ)=PIO(N)
    TG(IJ)=TEMIO(N)
    TH(IJ)=THIO(N)
    THN(IJ)=TH(IJ)
    UG(IJ)=UIO(N)
    VG(IJ)=VIO(N)
    DO 56 K=1,NPHASE
    IF(KIN(K).EQ.1)TSK(K,IJ)=TSKIO(K,N)
    TL(K,IJ)=TEMPIO(K,N)
    UK(K,IJ)=UPIO(K,N)
    VK(K,IJ)=VPPIO(K,N)
    RLK(K,IJ)=RL(K)*THPIO(K,N)
56  RLKN(K,IJ)=RLK(K,IJ)
    DO 55 K=1,LT
    DO 55 JX=1,JXN
    YN(K,JX,IJ)=YIO(K,JX,N)
55  Y(K,JX,IJ)=YIO(K,JX,N)
    CALL ROGY(IJ)
    RGP(IJ)=ROG(IJ)*TH(IJ)
60  CONTINUE
    ENDIF
    DO 65 J=2,JB1
    DO 65 I=2,IB1
    IJ=I+(J-1)*IB2
    IF((IFL(IJ).EQ.2).OR.(IFL(IJ).EQ.3))GOTO 65
    CALL INDXA
C  INITIAL BED PRESSURE

```

```

      IF(ITD.NE.2)THEN
      IF(IPRE.NE.0)THEN
      IF(IJ.EQ.IJB)THEN
      IF(IJ.NE.IJL)THEN
      CALL ROGY(IJL)
      CC=ROG(IJL)
      RLXSUM=TH(IJL)*CC
      DO 61 K=1,NPHASE
61    RLXSUM=RLXSUM+RLK(K,IJL)
      P(IJ)=P(IJL)+GRAVX*DR(I)*RLXSUM
      ENDIF
      ELSE
      CALL ROGY(IJB)
      CC=ROG(IJB)
      RLXSUM=TH(IJB)*CC
      DO 62 K=1,NPHASE
62    RLXSUM=RLXSUM+RLK(K,IJB)
      P(IJ)=P(IJB)+GRAVY*DZ(J)*RLXSUM
      ENDIF
      ENDIF
      ENDIF
C
      CALL CNVERT
      CALL MASFGA
      CALL MASFKA
      THN(IJ)=TH(IJ)
      TG(IJ)=C1+SIEG(IJ)/CG
      IF(IHEAT.EQ.0)TG(IJ)=TEMIO(1)
      CALL ROGY(IJ)
      DO 63 K=1,NPHASE
      IF (RLK(K,IJ).EQ.0.0) THEN
      TL(K,IJ)=0.0
      ELSE
      TL(K,IJ)=C2+SIEL(K,IJ)/CL(K)
      ENDIF
      IF(IHEAT.EQ.0)TL(K,IJ)=TEMPIO(K,1)
      CALL THRCON
      TSKN(K,IJ)=TSK(K,IJ)
      IF(KIN(K).EQ.1)CALL GRPROP
63    RLKN(K,IJ)=RLK(K,IJ)
      CALL KDRAGS
      CALL MULTI
      CALL RHEATS
65    CONTINUE
650   FORMAT(1X,78I1)

```


C -----SIELF

SUBROUTINE SIELF
INCLUDE 'mfrek.com'

C
C CALCULATES FLUXES OF SPECIFIC INTERNAL
C ENERGY DENSITY OF THE LIQ.
C

IF(UK(K,IJ).GE.0.)THEN
ELFR(K)=RLK(K,IJ)*SIELN(K,IJ)*UK(K,IJ)*RB(I)
ELSE
ELFR(K)=RLK(K,IJR)*SIELN(K,IJR)*UK(K,IJ)*RB(I)
ENDIF

IF(VK(K,IJ).GE.0.)THEN
ELFT(K)=RLK(K,IJ)*SIELN(K,IJ)*VK(K,IJ)
ELSE
ELFT(K)=RLK(K,IJT)*SIELN(K,IJT)*VK(K,IJ)
ENDIF

IF(IFL(IMJ).NE.1)GOTO 1
IF(IFL(IJM).NE.1)GOTO 2
CALL HEATLB
GOTO 4

1 IF(UK(K,IMJ).GE.0.)THEN
ELFL(K)=RLK(K,IJL)*SIELN(K,IJL)*UK(K,IMJ)*RB(I-1)
ELSE
ELFL(K)=RLK(K,IJ)*SIELN(K,IJ)*UK(K,IMJ)*RB(I-1)
ENDIF

IF(IFL(IJM).NE.1)GOTO 2
CALL HEATLA
GOTO 3

2 IF(VK(K,IJM).GE.0.)THEN
ELFB(K,I)=RLK(K,IJB)*SIELN(K,IJB)*VK(K,IJM)
ELSE
ELFB(K,I)=RLK(K,IJ)*SIELN(K,IJ)*VK(K,IJM)
ENDIF

CALL HEATCL
ELFB(K,I)=ELFB(K,I)-HFLB(K,I)
IF(IFL(IMJ).EQ.1)GOTO 4

3 ELFL(K)=ELFL(K)-HFLK(K)

4 ELFT(K)=ELFT(K)-HFLT(K)

ELFR(K)=ELFR(K)-HFLR(K)

RETURN

END

C -----SOLUTION

C NOT IN USE

SUBROUTINE SOLUTION(DK,DTO,C1,C2,C3,X)

```

      A=C1
      IF(C1.GE.2*C2)A=C2/2.0
      B=-C3
C     UNDERFLOW
      COEM=1.0/(C3+A)
      FA=COEM*DK*(C1-A)*(C2-2*A)*(C2-2*A)-
      1COEM*(C3+A)*(DTO-2*A)*(DTO-2*A)
      FB=COEM*DK*(C1-B)*(C2-2*B)*(C2-2*B)-
      1COEM*(C3+B)*(DTO-2*B)*(DTO-2*B)
1     X=(A+B)/2.0
      FX=COEM*DK*(C1-X)*(C2-2*X)*(C2-2*X)-
      1COEM*(C3+X)*(DTO-2*X)*(DTO-2*X)
      IF(ABS(FX/COEM).LE.1D-20)GOTO 10
      IF(FX.LT.0.0)THEN
        A=X
      ELSE
        B=X
      ENDIF
      GOTO 1
10    RETURN
      END

C -----TAPERD
      SUBROUTINE TAPERD
      INCLUDE 'mfrek.com'
C
      READ(9)TIME,TTIM
      READ(9) (P(IJ),TH(IJ),UG(IJ),VG(IJ),TG(IJ),IJ=1,IB2JB2)
      READ(9)((RLK(K,IJ),UK(K,IJ),VK(K,IJ),TL(K,IJ),
      1TSK(K,IJ),TSKN(K,IJ),K=1,NPHASE),IJ=1,IB2JB2)
      READ(9)((Y(K,JX,IJ),K=1,LT),JX=1,JXN),IJ=1,IB2JB2)
      READ(9)((YN(K,JX,IJ),K=1,LT),JX=1,JXN),IJ=1,IB2JB2)
      RETURN
      END

C -----TAPEWR
      SUBROUTINE TAPEWR
      INCLUDE 'mfrek.com'
      REWIND(9)
C
      WRITE(9)TIME,TTIM
      WRITE(9) (P(IJ),TH(IJ),UG(IJ),VG(IJ),TG(IJ),IJ=1,IB2JB2)
      WRITE(9)((RLK(K,IJ),UK(K,IJ),VK(K,IJ),TL(K,IJ),
      1TSK(K,IJ),TSKN(K,IJ),K=1,NPHASE),IJ=1,IB2JB2)
      WRITE(9)((Y(K,JX,IJ),K=1,LT),JX=1,JXN),IJ=1,IB2JB2)
      WRITE(9)((YN(K,JX,IJ),K=1,LT),JX=1,JXN),IJ=1,IB2JB2)
C     WRITE(6,*)'TIME=',TIME

```

RETURN
END

C -----THRCON

SUBROUTINE THRCON
INCLUDE 'mfrek.com'
PARAMETER (PIC=7.26D-3,RKP=0.3289,RKG=0.0269)
PARAMETER (APIC=1.-PIC,RF=RKP/RKG)

C

C DAMKOHLER'S EQUIVALENT THERMAL CONDUCTIVITY

C

IF (TH(IJ).LT.0.99) THEN
BB=1.25*((1.0-TH(IJ))/TH(IJ))**(10./9.)
CC=1.0-BB/RF
RKO=(2./CC)*(BB*(RF-1.)/(CC*CC*RF)*DLOG(RF/BB)-(BB-1.)/CC
1-(BB+1.)/2.)
A3=SQRT(1.0-TH(IJ))
AKL(K,IJ)=(1.-A3)+A3*(PIC*RF+APIC*RKO)*AKG(IJ)
ELSEIF (TH(IJ).EQ.1.0) THEN
AKL(K,IJ)=0.0
ELSE
AKL(K,IJ)=AKG(IJ)
ENDIF
RETURN
END

C -----TILDE

SUBROUTINE TILDE
INCLUDE 'mfrek.com'

C

DO 10 J=2,JB1
DO 10 I=2,IB1
IJ=I+(J-1)*IB2
IF (IFL(IJ).NE.1) GOTO 10
CALL INDX
CALL UGMOMF
IGKU=(AR(I)*TH(IJ)+BR(I)*TH(IJR))*1000
IGKV=(AZ(J)*TH(IJ)+BZ(J)*TH(IJT))*1000
IGJ=TH(IJ)*1000
IGT=TH(IJT)*1000
IGR=TH(IJR)*1000
RUG(IJ)=(AR(I)*RGP(IJ)+BR(I)*RGP(IJR))*(UG(IJ)+GRAVX*DT)
1-DTOBDR(I)*(UGFR-UGFL)-DTODZ(J)*(UGFT-UGFB(I))-DT*SUGC*ITC
UGFL=UGFR
UGFB(I)=UGFT
CALL VGMOMF
RVG(IJ)=(AZ(J)*RGP(IJ)+BZ(J)*RGP(IJT))*(VG(IJ)+GRAVY*DT)

```

      DO 20 I=2,IB1
      IJ=I+(J-1)*IB2
      IF(IFL(IJ).NE.1) GOTO 20
      CALL INDXA
      CALL MATSA
      CALL VELSK2
      CALL MASFGA
      CALL MASFKA
20    CONTINUE
      RETURN
      END

C -----UGMOMF
      SUBROUTINE UGMOMF
      INCLUDE 'mfrek.com'
C
      CS=0.5*(UG(IJ)+UG(IPJ))
      IF (CS.GE.0.) THEN
      UGFR=(AR(I)*RGP(IJ)+BR(I)*RGP(IJR))*UG(IJ)*CS*R(I+1)
      ELSE
      UGFR=(AR(I+1)*RGP(IJR)+BR(I+1)*RGP(IJRR))*UG(IPJ)*CS*R(I+1)
      ENDIF
      CS=AR(I)*VG(IJ)+BR(I)*VG(IPJ)
      IF (CS.GE.0.) THEN
      UGFT=(AR(I)*RGP(IJ)+BR(I)*RGP(IJR))*UG(IJ)*CS
      ELSE
      UGFT=(AR(I)*RGP(IJT)+BR(I)*RGP(IJTR))*UG(IPJ)*CS
      ENDIF
      IF(IFL(IMJ).NE.1) GOTO 1
      IF(IFL(IJM).NE.1) GOTO 2
      CALL UGVSB
      GOTO 4
1     CS=0.5*(UG(IJ)+UG(IMJ))
      IF (CS.GE.0.) THEN
      UGFL=(BR(I-1)*RGP(IJ)+AR(I-1)*RGP(IJL))*UG(IMJ)*CS*R(I)
      ELSE
      UGFL=(AR(I)*RGP(IJ)+BR(I)*RGP(IJR))*UG(IJ)*CS*R(I)
      ENDIF
      IF(IFL(IJM).NE.1) GOTO 2
      CALL UGVSA
      GOTO 3
2     CS=(AR(I)*VG(IJM)+BR(I)*VG(IPJM))
      IF (CS.GE.0.) THEN
      UGFB(I)=(AR(I)*RGP(IJB)+BR(I)*RGP(IJBR))*UG(IJM)*CS
      ELSE
      UGFB(I)=(AR(I)*RGP(IJ)+BR(I)*RGP(IJR))*UG(IJ)*CS

```

```

ENDIF
CALL UGVS
UGFB(I)=UGFB(I)-SUGB
IF(IFL(IMJ).EQ.1)GOTO 4
3  UGFL=UGFL-SUGL
4  UGFT=UGFT-SUGT
   UGFR=UGFR-SUGR
   RETURN
   END
C -----UGVS
   SUBROUTINE UGVS
   INCLUDE 'mfrek.com'
C
   SUGB=VISF*((VG(IPJM)-VG(IJM))*RDRP(I)+(UG(IJ)-UG(IJM))
   1*RDZP(J-1))*
   1(BZ(J-1)*(AR(I)*TH(IJ)+BR(I)*TH(IJR))
   1+AZ(J-1)*(AR(I)*TH(IJB)+BR(I)*TH(IJBR)))
C
   ENTRY UGVSA
   SUGL=VISF*(2.*(UG(IJ)-UG(IMJ))*RDR(I)-(2./3.)
   1*(RRIDR(I)*(RB(I)*
   1UG(IJ)-RB(I-1)*UG(IMJ))+(VG(IJ)-VG(IJM))
   1*RDZ(J)))*TH(IJ)*R(I)
C
   ENTRY UGVSB
   SUGT=VISF*((VG(IPJ)-VG(IJ))*RDRP(I)+(UG(IPJ)-
   1UG(IJ))*RDZP(J))*
   1(AZ(J)*(AR(I)*TH(IJ)+BR(I)*TH(IJR))
   1+BZ(J)*(AR(I)*TH(IJT)+BR(I)*TH(IJTR)))
   SUGR=VISF*(2.*(UG(IPJ)-UG(IJ))*RDR(I+1)-(2./3.)*(RRIDR(I+1)
   1*(RB(I+1)*
   1UG(IPJ)-RB(I)*UG(IJ))+(VG(IPJ)-VG(IPJM))*RDZ(J))
   1*TH(IJR)*R(I+1)
   SUGC=VISF*(2.*RRB(I)*UG(IJ)-(2./3.)*(0.5*RRIDRP(I)*
   1(RB(I+1)*UG(IPJ)-RB(I-1)*UG(IMJ))+
   1(AR(I)*(VG(IJ)-VG(IJM))
   1+BR(I)*(VG(IPJ)-VG(IPJM))*RDZ(J)))*RRB(I)
   1*(AR(I)*TH(IJ)+BR(I)*TH(IJR))
   RETURN
   END
C -----ULMOMF
   SUBROUTINE ULMOMF
   INCLUDE 'mfrek.com'
C
   CS=0.5*(UK(K,IJ)+UK(K,IPJ))

```

```

      IF (CS.GE.0.) THEN
        ULFR(K)=(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
        1*UK(K,IJ)*CS*R(I+1)
      ELSE
        ULFR(K)=(AR(I+1)*RLK(K,IJR)+BR(I+1)*RLK(K,IJRR))
        1*UK(K,IPJ)*CS*R(I+1)
      ENDIF
      CS=AR(I)*VK(K,IJ)+BR(I)*VK(K,IPJ)
      IF (CS.GE.0.) THEN
        ULFT(K)=(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))*UK(K,IJ)*CS
      ELSE
        ULFT(K)=(AR(I)*RLK(K,IJT)+BR(I)*RLK(K,IJTR))*UK(K,IJP)*CS
      ENDIF
      IF(IFL(IMJ).NE.1) GOTO 1
      IF(IFL(IJM).NE.1) GOTO 2
      CALL ULVSB
      GOTO 4
1     CS=0.5*(UK(K,IJ)+UK(K,IMJ))
      IF (CS.GE.0.) THEN
        ULFL(K)=(BR(I-1)*RLK(K,IJ)+AR(I-1)*RLK(K,IJL))
        1*UK(K,IMJ)*CS*R(I)
      ELSE
        ULFL(K)=(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))*UK(K,IJ)*CS*R(I)
      ENDIF
      IF(IFL(IJM).NE.1)GOTO2
      CALL ULVSA
      GOTO 3
2     CS=AR(I)*VK(K,IJM)+BR(I)*VK(K,IPJM)
      IF (CS.GE.0.) THEN
        ULFB(K,I)=(AR(I)*RLK(K,IJB)+BR(I)*RLK(K,IJBR))*UK(K,IJM)*CS
      ELSE
        ULFB(K,I)=(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))*UK(K,IJ)*CS
      ENDIF
      CALL ULVS
      ULFB(K,I)=ULFB(K,I)-SULB(K)
      IF(IFL(IMJ).EQ.1)GOTO 4
3     ULFL(K)=ULFL(K)-SULL(K)
4     ULFT(K)=ULFT(K)-SULT(K)
      ULFR(K)=ULFR(K)-SULR(K)
      RETURN
      END
C -----ULVS
      SUBROUTINE ULVS
      INCLUDE 'mfrek.com'
C

```

```

SULB(K)=((VK(K,IPJM)-VK(K,IJM))*RDRP(I)+(UK(K,IJ)-UK(K,IJM))
1*RDZP(J-1))*((BZ(J-1)*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1+AZ(J-1)*(AR(I)*RLK(K,IJB)+BR(I)*RLK(K,IJBR)))
1*ARL(K)*(VISSD(K)))

```

C

```

ENTRY ULVSA
SULL(K)=2.*(UK(K,IJ)-UK(K,IMJ))*RDR(I)*R(I)*RLK(K,IJ)*ARL(K)
1*VISSD(K)+(RRIDR(I)*(RB(I)
1*UK(K,IJ)-RB(I-1)*UK(K,IMJ)))+(VK(K,IJ)-VK(K,IJM))
1*RDZ(J))*R(I)
1*(VISBL(K,IJ)-RLK(K,IJ)*ARL(K)*(2./3.)*VISSD(K))

```

C

```

ENTRY ULVSB
SULT(K)=((VK(K,IPJ)-VK(K,IJ))*RDRP(I)+(UK(K,IJP)-UK(K,IJ))
1*RDZP(J))
1*((AZ(J)*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1+BZ(J)*(AR(I)*RLK(K,IJT)+BR(I)*RLK(K,IJTR)))*ARL(K)*(
1*VISSD(K)))
SULR(K)=2.*(UK(K,IPJ)-UK(K,IJ))*RDR(I+1)*R(I+1)*
1*RLK(K,IJR)*ARL(K)*(VISSD(K))+((RRIDR(I+1)
1*(RB(I+1)*UK(K,IPJ)-RB(I)*UK(K,IJ)))+(VK(K,IPJ)-VK(K,IPJM))
1*RDZ(J))*R(I+1)*(VISBL(K,IJR)-(2./3.)*RLK(K,IJR)*ARL(K)
1*(VISSD(K)))
SULC(K)=(2.*RRB(I)*UK(K,IJ)*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1*ARL(K)*(VISSD(K)))+(0.5*RRIDR(I)
1*(RB(I+1)*UK(K,IPJ)-RB(I-1)*UK(K,IMJ))
1+(AR(I)*(VK(K,IJ)-VK(K,IJM))
1+BR(I)*(VK(K,IPJ)-VK(K,IPJM)))*RDZ(J))*RRB(I)*
1*(VISBL(K,IJ)-(2./3.)*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1*ARL(K)*(VISSD(K)))
RETURN
END

```

C -----VELINV

```

SUBROUTINE VELINV(NH,NP,A,B)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(NP,NP),B(NP)

```

C

C USE GAUSS-DOLITTLE METHOD FOR SYMMETRIC MATRIX INVERSION

```

DO 136 K=2,NP
IF(ABS(A(K,K)).GE.1.D-6)GOTO 136
DO 135 KK=1,NP
A(K,KK)=0.0
135 A(KK,K)=0.0
B(K)=0.0
136 CONTINUE

```

C

```

DO 160 K=1,NP
IF(A(K,K).EQ.0.0)GOTO 160
KP1=K+1
DIV=1./A(K,K)
DO 140 KJ=KP1,NP
A(K,KJ)=A(K,KJ)*DIV
140 CONTINUE
B(K)=B(K)*DIV
DO 150 KI=KP1,NP
AMUL=A(KI,K)
DO 145 KJ=KP1,NP
145 A(KI,KJ)=A(KI,KJ)-AMUL*A(K,KJ)
150 B(KI)=B(KI)-AMUL*B(K)
160 CONTINUE
DO 170 K=NH,1,-1
KP1=K+1
DO 170 KI=KP1,NP
170 B(K)=B(K)-B(KI)*A(K,KI)
RETURN
END

```

C

-----VELSK

```

SUBROUTINE VELSK
INCLUDE 'mfrek.com'

```

C

```

IFLL=IFL(IMJ)
IF((IFLL.EQ.2).OR.(IFLL.EQ.3).OR.(IFLL.EQ.5))GOTO 200
CALL VELINV(NPHASE,NPHS1,AU1,BU1)
UG(IMJ)=BU1(1)
DO 165 K=2,NPHS1
165 UK(K-1,IMJ)=BU1(K)
200 CONTINUE
IFLB=IFL(IJM)
IF((IFLB.EQ.2).OR.(IFLB.EQ.3).OR.(IFLB.EQ.5))GOTO 300

```

C

```

CALL VELINV(NPHASE,NPHS1,AV1,BV1)
VG(IJM)=BV1(1)
DO 265 K=2,NPHS1
265 VK(K-1,IJM)=BV1(K)

```

C

```

ENTRY VELSK2
300 CONTINUE
IFLR=IFL(IPJ)
IF((IFLR.EQ.2).OR.(IFLR.EQ.3).OR.(IFLR.EQ.5))GOTO 400
CALL VELINV(NPHASE,NPHS1,AU,BU)

```



```

      UG(IJ)=BU(1)
      DO 365 K=2,NPHS1
365   UK(K-1,IJ)=BU(K)
400   CONTINUE
      IFLT=IFL(IJP)
      IF((IFLT.EQ.2).OR.(IFLT.EQ.3).OR.(IFLT.EQ.5))RETURN
      CALL VELINV(NPHASE,NPHS1,AV,BV)
      VG(IJ)=BV(1)
      DO 465 K=2,NPHS1
465   VK(K-1,IJ)=BV(K)
      RETURN
      END

```

C -----VGMOMF

```

      SUBROUTINE VGMOMF
      INCLUDE 'mfrek.com'

```

```

C
      CS=0.5*(VG(IJ)+VG(IJP))
      IF (CS.GE.0.) THEN
        VGFT=(AZ(J)*RGP(IJ)+BZ(J)*RGP(IJT))*VG(IJ)*CS
      ELSE
        VGFT=(AZ(J+1)*RGP(IJT)+BZ(J+1)*RGP(IJTT))*VG(IJP)*CS
      ENDIF
      CS=AZ(J)*UG(IJ)+BZ(J)*UG(IJP)
      IF (CS.GE.0.) THEN
        VGFR=(AZ(J)*RGP(IJ)+BZ(J)*RGP(IJT))*VG(IJ)*CS*RB(I)
      ELSE
        VGFR=(AZ(J)*RGP(IJR)+BZ(J)*RGP(IJTR))*VG(IJP)*CS*RB(I)
      ENDIF
      IF(IFL(IMJ).NE.1) GOTO 1
      IF(IFL(IJM).NE.1) GOTO 2
      CALL VGVSB
      GOTO 4
1     CS=AZ(J)*UG(IMJ)+BZ(J)*UG(IMJP)
      IF (CS.GE.0.) THEN
        VGFL=(AZ(J)*RGP(IJL)+BZ(J)*RGP(IJTL))*VG(IMJ)*CS*RB(I-1)
      ELSE
        VGFL=(AZ(J)*RGP(IJ)+BZ(J)*RGP(IJT))*VG(IJ)*CS*RB(I-1)
      ENDIF
      IF(IFL(IJM).NE.1)GOTO2
      CALL VGVSA
      GOTO 3
2     CS=0.5*(VG(IJM)+VG(IJ))
      IF (CS.GE.0.) THEN
        VGFB(I)=(BZ(J-1)*RGP(IJ)+AZ(J-1)*RGP(IJB))*VG(IJM)*CS
      ELSE

```

```

VGFB(I)=(AZ(J)*RGP(IJ)+BZ(J)*RGP(IJT))*VG(IJ)*CS
ENDIF
CALL VGVVS
VGFB(I)=VGFB(I)-SVGB
IF(IFL(IMJ).EQ.1)GOTO 4
3  VGFL=VGFL-SVGL
4  VGFT=VGFT-SVGT
   VGFR=VGFR-SVGR
   RETURN
   END
C -----VGVS
   SUBROUTINE VGVVS
   INCLUDE 'mfrek.com'
C
   SVGB=VISF*(2./3.)*(2.*(VG(IJ)-VG(IJM))*RDZ(J)
   1-RRIDR(I)*(RB(I)*
   1UG(IJ)-RB(I-1)*UG(IMJ)))*TH(IJ)
C
   ENTRY VGVSA
   SVGL=VISF*((VG(IJ)-VG(IMJ))*RDRP(I-1)+(UG(IMJP)-UG(IMJ))
   1*RDZP(J))*
   1RB(I-1)*(AZ(J)*(BR(I-1)*TH(IJ)+AR(I-1)*TH(IJL))
   1+BZ(J)*(BR(I-1)*TH(IJT)+AR(I-1)*TH(IJTL)))
C
   ENTRY VGVSB
   SVGT=VISF*(2./3.)*(2.*(VG(IJP)-VG(IJ))*RDZ(J+1)-RRIDR(I)
   1*(RB(I)*
   1UG(IJP)-RB(I-1)*UG(IMJP)))*TH(IJT)
   SVGR=VISF*((VG(IJP)-VG(IJ))*RDRP(I)+(UG(IJP)-UG(IJ))
   1*RDZP(J))*
   1RB(I)*(AZ(J)*(AR(I)*TH(IJ)+BR(I)*TH(IJR))
   1+BZ(J)*(AR(I)*TH(IJT)+BR(I)*TH(IJTR)))
   RETURN
   END
C -----VLMOMF
   SUBROUTINE VLMOMF
   INCLUDE 'mfrek.com'
C
   CS=0.5*(VK(K,IJ)+VK(K,IJP))
   IF (CS.GE.0.) THEN
   VLFT(K)=(AZ(J)*RLK(K,IJ)+BZ(J)*RLK(K,IJT))*VK(K,IJ)*CS
   ELSE
   VLFT(K)=(AZ(J+1)*RLK(K,IJT)+BZ(J+1)*RLK(K,IJTT))
   1*VK(K,IJP)*CS
   ENDIF

```

```

CS=AZ(J)*UK(K,IJ)+BZ(J)*UK(K,IJP)
IF (CS.GE.0.) THEN
VLFR(K)=(AZ(J)*RLK(K,IJ)+BZ(J)*RLK(K,IJT))*VK(K,IJ)*CS*RB(I)
ELSE
VLFR(K)=(AZ(J)*RLK(K,IJR)+BZ(J)*RLK(K,IJTR))
1*VK(K,IJP)*CS*RB(I)
ENDIF
IF(IFL(IMJ).NE.1) GOTO 1
IF(IFL(IJM).NE.1) GOTO 2
CALL VLVSB
GOTO 4
1 CS=AZ(J)*UK(K,IMJ)+BZ(J)*UK(K,IMJP)
IF (CS.GE.0.) THEN
VLFL(K)=(AZ(J)*RLK(K,IJL)+BZ(J)*RLK(K,IJTL))
1*VK(K,IMJ)*CS*RB(I-1)
ELSE
VLFL(K)=(AZ(J)*RLK(K,IJ)+BZ(J)*RLK(K,IJT))
1*VK(K,IJ)*CS*RB(I-1)
ENDIF
IF(IFL(IJM).NE.1)GOTO2
CALL VLVSA
GOTO 3
2 CS=0.5*(VK(K,IJM)+VK(K,IJ))
IF (CS.GE.0.) THEN
VLFB(K,I)=(BZ(J-1)*RLK(K,IJ)+AZ(J-1)*RLK(K,IJB))
1*VK(K,IJM)*CS
ELSE
VLFB(K,I)=(AZ(J)*RLK(K,IJ)+BZ(J)*RLK(K,IJT))*VK(K,IJ)*CS
ENDIF
CALL VLVS
VLFB(K,I)=VLFB(K,I)-SVLB(K)
IF(IFL(IMJ).EQ.1)GOTO 4
3 VLFL(K)=VLFL(K)-SVLL(K)
4 VLFT(K)=VLFT(K)-SVLT(K)
VLFR(K)=VLFR(K)-SVLR(K)
RETURN
END
C -----VLVS
SUBROUTINE VLVS
INCLUDE 'mfrek.com'
C
SVLB(K)=(2./3.)*(2.*(VK(K,IJ)-VK(K,IJM))*RDZ(J)-RRIDR(I)
1*(RB(I)*UK(K,IJ)-RB(I-1)*UK(K,IMJ)))
1*(RLK(K,IJ)*ARL(K)*(VISSD(K)))
1+VISBL(K,IJ)*RRIDR(I)*(RB(I)*UK(K,IJ)-RB(I-1)*UK(K,IMJ))

```

C

```

ENTRY VLVSA
SVLL(K)=((VK(K,IJ)-VK(K,IMJ))*RDRP(I-1)
1+(UK(K,IMJP)-UK(K,IMJ))
1*RDZP(J))*RB(I-1)*((AZ(J)*(BR(I-1)*RLK(K,IJ)+AR(I-1)
1*RLK(K,IJL))
1+BZ(J)*(BR(I-1)*RLK(K,IJT)+AR(I-1)*RLK(K,IJTL)))
1*ARL(K)*(VISSD(K)))
ENTRY VLVSB
SVLT(K)=(2./3.)*(2.*(VK(K,IJP)-VK(K,IJ))*RDZ(J+1)-RRIDR(I)
1*(RB(I)*UK(K,IJP)-RB(I-1)*UK(K,IMJP)))
1*(RLK(K,IJT)*ARL(K)*(VISSD(K)))
1+VISBL(K,IJT)*RRIDR(I)*(RB(I)*UK(K,IJP)-RB(I-1)*UK(K,IMJP))
SVLR(K)=((VK(K,IPJ)-VK(K,IJ))*RDRP(I)+(UK(K,IJP)-UK(K,IJ))
1*RDZP(J))*RB(I)*((AZ(J)*(AR(I)*RLK(K,IJ)+BR(I)*RLK(K,IJR))
1+BZ(J)*(AR(I)*RLK(K,IJT)+BR(I)*RLK(K,IJTR)))
1*ARL(K)*(VISSD(K)))
RETURN
END

```

C -----VWORKL

```

SUBROUTINE VWORKL
INCLUDE 'mfrek.com'

```

C

C CALCULATE VISCOUS STRESSES FOR THE WORK TERM IN THE PHASES
C ENERGY EQUATION (NOTE: VISCOSITY IS MULTIPLIED LATER)

C

```

DO 100 J=2,JB1
DO 100 I=2,IB1
IJ=I+(J-1)*IB2
IF(IFL(IJ).EQ.1)THEN
CALL INDX
C CALL OBSTACLE
DO 10 K=1,NPHASE
IF(KIN(K).NE.1)GOTO 10
IF(IFL(IPJM).EQ.3)VK(K,IPJM)=0.0
IF(IFL(IMJM).EQ.3)VK(K,IMJM)=0.0
IF(IFL(IMJP).EQ.3)UK(K,IMJP)=0.0
IF(IFL(IMJM).EQ.3)UK(K,IMJM)=0.0
CS1=0.5*((VK(K,IPJ)+VK(K,IPJM)-VK(K,IMJ)-VK(K,IMJM))
1/(DRP(I)+DRP(I-1)))+(RB(I)*(UK(K,IJP)-UK(K,IJM))
1+RB(I-1)*(UK(K,IMJP)-UK(K,IMJM)))
1/(R(I)*(DZP(J)+DZP(J-1))))
CS2=RDZ(J)*(VK(K,IJ)-VK(K,IJM))
CS3=RDR(I)*(UK(K,IJ)-UK(K,IMJ))
CS4=0.5*(RB(I)*UK(K,IJ)+RB(I-1)*UK(K,IMJ))/(R(I)*R(I))

```

C

```

SILM(K,IJ)=RRIDR(I)*(RB(I)*UK(K,IJ)
1-RB(I-1)*UK(K,IMJ))+CS2
SILRZ=CS1
SILZZ=2.0*CS2
SILRR=2.0*CS3
SILPP=2.0*CS4

```

C

C CALCULATE VISCOUS WORK FOR THE PHASES

```

VWLS(K,IJ)=DT*(SILRZ*CS1+SILZZ*CS2+SILRR*CS3
1+SILPP*CS4*ITC)
VWLM(K,IJ)=DT*SILM(K,IJ)*SILM(K,IJ)
10 CONTINUE
ENDIF
100 CONTINUE
RETURN
END

```

C.2 MFREK.COM. The followings are common file for "mfrek.f" and its name must be "mfrek.com" and in same subdirectory as that "mfrek.f" is.

```

IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER (IB2=18,JB2=60,NPHASE=2,NPHS1=NPHASE+1,LT=NPHS1)
PARAMETER (NIN=1,NOUT=1,NFL=3,NOBS=6,JXN=6,THICK=44.8564)
PARAMETER (ITC=0,WMYGAS=28.5,THMIN=0.36,CRES=0.99999)
PARAMETER (CG=1.80041D7,VISF=1.82D-4)
PARAMETER (RST=0.,IPRE=0)
PARAMETER (C17=8.314D7,C18=0.0 )
PARAMETER (IB=IB2-2,IB1=IB2-1,JB=JB2-2,JB1=JB2-1)
PARAMETER (IB3=IB2+1,IB4=IB2+2,IB2JB2=IB2*JB2)
PARAMETER (IB2JB1=IB2JB2-IB2,IB1JB2=IB2JB2-1,
1IB2JB0=JB*IB2+1)
PARAMETER (NO=NIN+NOUT+NFL,NT=NO+NOBS)
PARAMETER (NH=NPHASE,NC=IB2JB2,NI=IB2,NJ=JB2,
1 NP=NPHS1,NF=NP*(NP+1)/2)
PARAMETER (PI=3.14159265359)

```

C SETC

```

PARAMETER (C1=300.,C2=300)

```

C COMMON BLOCKS

```

COMMON / PARAM1 / DT,KIN(NH),IRXN,IHEAT,VISS(NH),
1 DG, D1, D2, D3, P1, P2, P3, GRAVX,GRAVY,NIT,
1 TIME, TDUMP, TPR, TSTOP, TARRAY(2), TTIM
COMMON / PARAM2 / ITHMF(NT), QF(NC),
1 AYM(JXN,JB2),AYMOL(JXN,JB2),AYMOL1(JXN,JB2),YMOL(JXN,NC),
1 FLUX(JB2),AMIX(20),

```

```

1 TG(NC), TL(NH,NC), TGN(NC), TLN(NH,NC),
1 SIEG(NC), SIEGN(NC), AKG(NC),
1 SIEL(NH,NC), SIELN(NH,NC), AKL(NH,NC),
1 CL(NH), RHEAT(NH,NC),
1 UGFL, UGFR, UGFT, UGFB(NI),
1 ULFL(NH), ULFR(NH), ULFT(NH), ULFB(NH,NI),
1 VGFL, VGFR, VGFT, VGFB(NI),
1 VLFL(NH), VLFR(NH), VLFT(NH), VLFB(NH,NI),
1 THFL, THFR, THFT, THFB(NI),
1 EGFL, EGFR, EGFT, EGFB(NI),
1 HFGL, HFGR, HFGT, HFGB,
1 ELFL(NH), ELFR(NH), ELFT(NH), ELFB(NH,NI),
1 HFL(NH), HFLR(NH), HFLT(NH), HFLB(NH,NI),
1 OMTFL(NH), OMTFR(NH), OMTFT(NH), OMTFB(NH,NI),
1 SUGL, SUGR, SUGT, SUGB, SUGC,
1 SULL(NH), SULR(NH),SULT(NH), SULB(NH), SULC(NH),
1 SVGL, SVGR, SVGT, SVGB,
1 SVLL(NH), SVLR(NH),SVLT(NH), SVLB(NH)
COMMON / PARAM3 /
1 IOB(4,NT), ARL(NH), DK(NH), GTH(0:1000), DKF(NH,NH),
1 C(16), RL(NH),VISSD(NH),IFL(NC),INDS(NC,5), PHI(NH),
1 CPHI(NH), PHILIM(NH,NH), EPSL(NH,NH), EPSU(NH,NH),
1 R(NI), RB(NI), RRB(NI),COHF(0:1000),
1 DR(NI), DZ(NJ), DRP(NI), DZP(NJ),
1 RDR(NI), RDZ(NJ), RDRP(NI), RDZP(NJ),
1 AR(NI), BR(NI), AZ(NJ), BZ(NJ),
1 DTODR(NI), DTODZ(NJ), DTODRP(NI), DTODZP(NJ),
1 RRIDR(NI), RRIDRP(NI), DTOBDR(NI), DTORDR(NI)
COMMON / PARAMI /
1 I, IJ, IJB, IJBR, IJL, IJM, IJP, IJR,IMJM,
1 IJRR, IJT, IJTL, IJTR, IJTT, IMJ, IMJP, IPJ,
1 IPJM, IPJP, ITD, J, K, NSL(4), NSO(NT)
COMMON/AREA1/
1 ABETA(NC), CONV(NC), P(NC), PN(NC), TH(NC), THN(NC),
1 RGFR(NC), RGFT(NC), RGP(NC),RGPN(NC), ROG(NC),
1 RGFRY(NC),RGFTY(NC),
1 APP(NF,NC), RUG(NC), RVG(NC),
1 AU1(NP,NP), AV1(NP,NP), AU(NP,NP), AV(NP,NP),
1 BU1(NP), BV1(NP), BU(NP), BV(NP)
COMMON/AREA2/
1 RLFRK(NH,NC), RLFTK(NH,NC), RKPG(NH,NC),
1 RLFRKY(NH,NC),RLFTKY(NH,NC),
1 RUK(NH,NC), RVK(NH,NC), RLK(NH,NC), RLKN(NH,NC),
1 UG(NC), VG(NC), UK(NH,NC), VK(NH,NC)
COMMON/INPUT/

```

```

1 UIO(NO), VIO(NO), PIO(NO), THIO(NO), TEMIO(NO),
1 UPIO(NH,NO), VPIO(NH,NO), THPIO(NH,NO), TEMPIO(NH,NO),
1 WM(JXN),Q(NC),DOTM(LT),DOTMJ(LT,JXN),YIO(LT,JXN,NO),
1 QGEN(LT),QFLOW(LT),IHO(LT),IHE(LT),HHO(LT,5),
1 HHE(LT,5),AHO(LT,5,JXN),AHE(LT,5,JXN),Y(LT,JXN,NC),
1 RHO(LT,5),RHE(LT,5),TOB(NC),YN(LT,JXN,NC),
1 KEY(4),AHK(LT,5,JXN),DGG(NC),QQ(NT),TOBB(NT),COEK(NT),
1 COE(NC),
1 RLKMIN(NH),TSKIO(NH,NO),TSK(NH,NC),VISBL(NH,NC),
1 VISCL(NH,NC),VISDIL(NH),PS(NH,NC),TSKN(NH,NC),
1 SILM(NH,NC),VWLM(NH,NC),VWLS(NH,NC),GCON(NH,NC),
1 GAMMA(NH,NC),TSKCB(NH),TSKFB(NH,NI),THL(NH,NC),
1 TSKCL(NH),TSKCT(NH),TSKCR(NH),
1 TSKFL(NH),TSKFT(NH),TSKFR(NH)

```

C.3 MFREK.D. The followings are a sample input data file named "mfrek.d".

mfrek.r

SAMPLE INPUT DATA

18*1.786

60*13.31

5	2	3	3		
1	0	2	17	2	13
1	0	2	17	14	31
1	0	2	17	32	59
5	0	2	17	1	1
7	0	2	17	60	60
3	1	12	13	40	42
0	273.15		2.0D6		
3	1	12	13	34	36
0	273.15		2.0D6		
3	1	12	13	28	30
0	273.15		2.0D6		
3	1	12	13	22	24
0	273.15		2.0D6		
3	1	12	13	16	18
0	273.15		2.0D6		
3	1	12	13	10	12
0	273.15		2.0D6		
0.005	0.1				
1.0	1.0				
3.011	0.70025				
4.9D7	3.0D7				
5.0	0.089				
1	0.1	0	0.1		

0	4.80	5.3129D7	0.050	523.15		
0	0		0.524	523.15	1.0	
0	0		0.426	523.15	1.0	
0	4.80	5.2944D7	0.050	523.15		
0	0		0	0	1.0	
0	0		0.950	523.15	1.0	
0	0.24	5.2944D7	1.000	523.15		
0	0		0	0	1.0	
0	0		0	0	1.0	
0	0.24	5.3349D7	1.000	523.15		
0	0		0	0	1.0	
0	0		0	0	1.0	
0	0.24	5.2944D7	1.000	523.15		
0	0		0	0	1.0	
0	0		0	0	1.0	
1	1					
28	44	2	16	32	28	
1	0					
0	-1	0	-2	0	1	0
0	0					
0	0					
0.6806		0.2726	0.0334	0.0000	0.0000	0.0134
0		0	0	0	0	0
0		0	0	0	0	0
0.6806		0.2726	0.0334	0.0000	0.0000	0.0134
0		0	0	0	0	0
0		0	0	0	0	0
0.6806		0.2726	0.0334	0.0000	0.0000	0.0134
0		0	0	0	0	0
0		0	0	0	0	0
0.6806		0.2726	0.0334	0.0000	0.0000	0.0134
0		0	0	0	0	0
0		0	0	0	0	0
0.6806		0.2726	0.0334	0.0000	0.0000	0.0134
0		0	0	0	0	0
0		0	0	0	0	0
0						
0	30	1D-4				
0.05	0.005					
0	-980.62					

BIBLIOGRAPHY

- Air Products and Chemicals, Inc., "Liquid-Entrained Cataly Operations at LaPorte Methanol Process 1984-1985," Final Report to DOE for Contract No. DE-AC-22-81PC300019, 1987.
- Air Products and Chemicals, Inc., "Liquid Phase Methanol LaPorte Process Development Unit: Modification, Operation, and Support Studies," Drft Report to DOE for Contract No. DE-AC22-87PC90005, 1991.
- Air Products and Chemicals, Inc. "Synthesis of Dimethyl Ether and Alternative fuels in the Liquid Phase from Coal-Derived Synthesis Gas," Topical Report to DOE for Contract No. DE-AC22-90PC89865, 1992.
- Alfano, G., Carta, M., Ciccu, R., Del Fa, C., Peretti, R. and Zucca, A., "Applications of Static Electricity in Coal and Ore Beneficiation," J. Electrostatics, Vol. 16, pp. 315-328, 1985.
- Arastoopour, H. and Gidaspow, D., "Vertical Pneumatic Conveying Using Four Hydrodynamic Models," I&EC Fundam., Vol. 18, pp. 123-130, 1979.
- Bahary, M., "Experimental and Computational Studies of Hydrodynamics in Three-Phase and Two-Phase Fluidized Beds," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1994.
- Bukur, D. B., Daly, J. G. and Patel, S. A., "Hydrodynamics of Three-Phase Slurry Fischer-Tropsch Bubble Column Reactors," Final Report to the Department of Energy for contract No. DE-AC22-86PC90012, 1990.
- Cebeci, T. and Smith, A. M. O., Analysis of Turbulent Boundary Layers, Academic Press, NY, 1974.
- Chapman, S. and Cowling, T. G., "The Mathematical Theory of Non-Uniform Gases," 3rd ed., Cambridge Univ. Press, 1970.
- Clark, K. N., "The Effect of High Pressure and Temperature on Phase Distributions in a Bubble Column," Chem. Eng. Sci., Vol. 45, No. 8, pp. 2301-2307, 1990.
- Courant, R., Isaacson, E. and Rees, M., "On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences," Communication in Pure and Applied Mathematics, 5, p. 243, 1952.

- Deckwer, W. D., Burckhart, R. and Zoll, G., "Mixing and Mass Transfer in Tall Bubble Columns," Chem. Eng. Sci., Vol. 29, pp. 2177-2188, 1974.
- Deckwer, W. D., Louisi, Y., Zaidi, A. and Ralek, M., "Hydrodynamic Properties of the Fischer-Tropsch Slurry Process," Ind. Eng. Process Des. Dev., Vol. 19, pp. 699-708, 1980.
- Ding, J. and Gidaspow, D., "A Bubbling Fluidization Model Using Kinetic Theory of Granular Flow," AIChE J., Vol. 36, pp. 523-538, 1990.
- Ding, J., "A Fluidization Model Using Kinetic Theory of Granular Flow," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1990
- Ettehadieh, B., "Hydrodynamic Analysis of Gas-Solid Fluidized Beds," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1982.
- Gamwo, I. K., "Multi-Dimensional Hydrodynamic and Erosion Modeling of Fluidized Beds," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1992.
- Gidaspow, D., "Fluid Particle Systems," Two-Phase Flow and Heat Transfer, edited by Kakac, S. and Mayinger, F., Hemisphere Publish Co., Washington, DC, 1977.
- Gidaspow, D. and Ettehadieh, B., "Fluidization in Two-Dimensional Beds with a Jet, Part II: Hydrodynamic Modeling," I&EC Fundam., Vol. 22, pp. 193-201, 1983a.
- Gidaspow, D., Lin, C. and Seo, Y. C., "Fluidization in Two-Dimensional Beds with a Jet, Part I: Experimental Porosity Distributions," I&EC Fundam., 22, pp. 187-193, 1983b.
- Gidaspow, D., Syamlal, M. and Seo, Y. C., "Hydrodynamics of Fluidization: Supercomputer Generated vs Experimental Bubbles," In proc. of the 10th Powder and Bulk Solids Conf., Rosemont, Illinois, pp. 111-117, 1985.
- Gidaspow, D., "Hydrodynamics of Fluidization and Heat Transfer: Supercomputer Modeling," Appl. Mech. Rev., 39, No. 1, 1986.
- Gidaspow, D., Gupta, R., Mukherjee, A. and Wasan, D. T., "Separation of Pyrites from Illinois Coals Using Electrofluidized Beds and Electrostatic Sieve Conveyors," Processing and Utilization of High Sulfur coals II, edited by Chugh, Y. P. and Caudle, R. D., 1987.
- Gidaspow, D., Shih, Y. T., Bouillard, J. and Wasan, D., "Hydrodynamics of a Lamella Electrosettler," AIChE J., Vol. 35, No. 5, pp. 714-724, 1989.

- Gidaspow, D., Jayaswal, U. K. and Ding, J., "Navier-Stokes Equation Model for Liquid-Solid Flows Using Kinetic Theory", Liquid Solid Flows, FED, Vol. 118, ASME, New York, pp. 165-172, 1991.
- Gidaspow, D., "Multiphase Flow and Fluidization--Continuum and Kinetic Theory Descriptions," Academic Press, Inc., 1994.
- Gidaspow, D. Bahary, M. and Wu, Y., "Hydrodynamic Models for Slurry Bubble Column Reactors," Coal Liquefaction and Gas Conversion Contractors Review Conference, Pittsburgh, PA, August, 1995.
- Graff, G.H., Winkelman, J.G.M., Stamhuis, E.J. and Beenackers, A.A.C.M., "Kinetics of the Three-Phase Methanol Synthesis," Chem. Eng. Sci., Vol. 43, No. 8, pp.2161-2168, 1988.
- Gupta, R., "Dry Electrostatic Beneficiation of Illinis Coal and Eastern Oil Shales," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1990.
- Gupta, R., Gidaspow, D. and Wasan, D. T., "Electrostatic Separation of Powder Mixtures Based on the Work Functions of Its Constituents," Powder Technology, 75, pp. 79-87, 1993.
- Hammer, H., Schrag, H. Hektor, K., Schonau, K., Kusters, W., Soemarno, A., Sahabi, U. and Napp, W., "New Sub function on Hydrodynamics, Heat and Mass Transfer for Gas/Liquid and Gas/Liquid/Solid Chemical and Biochemical Reactors," Front. Chem. React. Eng., pp 464-474, 1984.
- Idogawa, K., Ikeda, K. Fukuda, T. and Morooka, S., "Behavior of Bubbles of the Air-Water System in a Column Under High Pressure," Int. Chem. Eng., Vol. 26, No.3, pp. 468-474, 1986.
- Idogawa, K., Ikeda, K. Fukuda, T. and Morooka, S., "Effect of Gas and Liquid Properties on the Behavior of Bubbles in a Column Under High Pressure," Int. Chem. Eng., Vol. 27, No. 1, pp. 93-99, 1987.
- Jackson, R., "Hydrodynamic Stability of Fluid-Particle Systems," Fluidization, edited by Davidson, J. F., Clift, R. and Harrison, D., Academic Press, pp. 47-72, 1985.
- Jayaswal, U. K., "Hydrodynamics of Multiphase Flows, Separation, Dissemination and Fluidization," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1991.
- Jayaswal, U. K., Gidaspow, D. and Wasan, D. T., "Continuous Separation of Fine Particles in Nonaqueous Media in a Lamella Electrosettler," Separation Technology, Vol. 1, 1990.

- Jenkins, J. T. and Savage, S. B., "A Theory for the Rapid of Identical, Smooth, Nearly Elastic, Spherical Particles," J. Fluid Mech., 130, pp. 187-202, 1983.
- Koide, K., Takazama, A., Komura, M. and matsunaga, H., "Gas Holdup and Volumetric Liquid Phase Mass Transfer Coefficient in Solid-Suspended Bubble Columns," J. Chem. Eng. Jpn., Vol. 1, p. 459, 1984.
- Kojima, H., Okumura, B. and Nakamura, A., "Effect of Pressure on Gas Holdup in a Bubble Column and a Slurry Bubble Column," J. Chem. Eng. Japan, Vol. 24, No. 1, pp. 115-117, 1991.
- Ledakowicz, S., Nettelhoff, H. and Deckwer, W.D., "Gas-liquid Mass Transfer Data in a Stirred Autoclave Reactor," Ind. Engng. Chem. Fundam., 23, pp. 510-512, 1984.
- Lee, S., "Methanol Synthesis Technology," CRC Press Inc., 1990.
- Lyczkowski, R. W., "Characteristics and Stability Analyses of Transient One-Dimensional Two Phase Flow Equations and Their Finite Difference Approximations," Nuclear Science and Engineering, 66, pp. 378-396 (1978).
- Mukherjee, A., "Characterization and Separation of Charged Particles," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1997.
- Noye, B. J., "An Introduction to Finite Difference Techniques," Numerical Simulation of Fluid Motion, ed. Noye, J., North Holland Publ. Co., NY, pp. 1-112, 1978.
- Ozturk, S. S., Schumpe, A. and Deckwer, W. D., "Organic Liquids in a Bubble Column: Holdups and Mass Transfer Coefficients," AIChE J., Vol. 33, No. 9, pp. 1473-1480, 1987.
- Pape, R., Gidaspow, D. and Wu, S., "Multiphase Flow in Slurry Bubble Column Reactors and Solid Propellant Rockets," To be presented at the 2nd International ymposium on Numerical Methods for Multiphase Flows, ASME Fluids Engineering Division, San Diego, CA, July, 1996.
- Reilly, I. G., Scott, D. S., Bruijn, D. T., Jain, A. and Diskorz, J., "Correlation for Gas Holdup in Turbulent Coalescing Bubble Columns," Can. J. Chem. Eng., Vol. 64, pp. 705-717, 1986.
- Rivard, W. C. and Torrey, M. O., "K-FIX: A Computer Program for Transient, Two Dimensional, Two Fluid Flow," Los Alamos, LA-NUREG-6623, 1977.

- Sada, E., Kumazawa, H., Lee, C. and Iguchi, T., "Gas Holdup and Mass-Transfer Characteristics in a Three-Phase Bubble Column," Ind. Eng. Chem. Process Des. Dev., Vol.25, pp. 472-476, 1986.
- Sauer, T. and Hempel, D. C., "Fluid Dynamics and Mass Transfer in a Bubble Column with Suspended Particles," Chem. Eng. Technol., Vol. 10, p. 180, 1987.
- Savage, S. B. and Sayed, M., "Stresses Developed by Dry Cohesionless Granular Materials Sheared in an Annular Shear Cell," J. Fluid Mech., 142, pp. 391-430, 1984.
- Savage, S.B., "Granular Flows at High Shear Rates," Theory of Dispersed Multiphase Flow, edited by Meyer, R. E., Academic Press, pp. 339-358, 1983.
- Savage, S. B. and Jeffery, D. J., "The Stress Tensor in a Granular Flow at High Shear Rates," J. Fluid Mech., 110, pp. 225-272, 1981.
- Schumpe, A., Saxena, A. K. and Fang, L.K., "Gas/Liquid Mass Transfer in a Slurry Bubble Column," Chem. eng. Sci., Vol. 42, No. 7, pp. 1787-1796, 1987.
- Shih, Y. T., Gidaspow, D. and Wasan, D. T., "Hydrodynamics of Electrofluidization: Separation of Pyrites from Coal," AIChE J., Vol. 33, No. 8, pp. 1322-1333, 1987.
- Soo, S. L., "Fluid Dynamics of Multiphase Systems," Blaisdell Publish Co., Waltham, MA, 1967.
- Syamlal, M., "Multiphase Hydrodynamics of Gas-Solids Flow," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1985.
- Therdthianwong, A., "Hydrodynamics and SO₂ Sorption in a Circulating Fluidized Bed," Ph.D. Thesis, Illinois Institute of Technology, Chicago, Illinois, 1994.
- Viking Systems International, "Design of Slurry Reactor for Indirect Liquefaction Applications," Report to DOE/PETC, 1993.
- Wedel, W.V., Ledakowicz, S. and Deckwer, W.D., "Kinetics of Methanol Synthesis in the Slurry Phase," Chem. Eng. Sci., Vol.43, No. 8, pp. 2169-2174, 1988.
- Weimer, R. F., Terry, D. M., and Stepanoff, P. R., "Laboratory Kinetics and Mass Transfer in the Liquid Phase Methanol Process," AIChE National Meeting, New York, 1987.

- Wu, Y. and Gidaspow, D., "Hydrodynamic Models for Slurry Bubble Column Reactors," 1st Technical Progress Report to DOE for Contract No. DE-FG22-94PC94208, Oct., 1994.
- Wu, Y. and Gidaspow, D., "Hydrodynamic Models for Slurry Bubble Column Reactors," 3rd Technical Progress Report to DOE for Contract No. DE-FG22-94PC94208, April., 1995.
- Zheng, C., Yao, B. and Feng, Y., "Flow Regime Identification and Gas Holdup of Three-phase Fluidized Systems," Chem. Eng. Sci., Vol. 43, No. 8, pp. 2195-2200, 1988.