A

Project Report on

CFD Modeling of Hydrodynamics of Fluidized Bed

In partial fulfillment of the requirements of

Bachelor of Technology (Chemical Engineering)

Submitted By

Paramvir Ahlawat (Roll No.10500012)

Session: 2008-09

Department of Chemical Engineering
National Institute of Technology
Rourkela-769008
Orissa

A

Project Report on

CFD Modeling of Hydrodynamics of Fluidized Bed

In partial fulfillment of the requirements of

Bachelor of Technology (Chemical Engineering)

Submitted By

Paramvir Ahlawat (Roll No.10500012)

Session: 2008-09

Under the guidance of

Prof. (Dr.) B. Munshi

Department of Chemical Engineering
National Institute of Technology
Rourkela-769008
Orissa

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that that the work in this thesis report entitled "**CFD Modeling of Hydrodynamics of Fluidized Bed**" submitted by Paramvir Ahalwat in partial fulfillment of the requirements for the degree of Bachelor of Technology in Chemical Engineering Session 2005-2009 in the department of Chemical Engineering, National Institute of Technology Rourkela, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge the matter embodied in the thesis has not been submitted to any other University /Institute for the award of any degree.

Date:

Prof. (Dr) B. Munshi
Department of Chemical Engineering
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

I owe a debt of deepest gratitude to my thesis supervisor, **Dr. B. Munshi**, Professor, Department of Chemical Engineering, for his guidance, support, motivation and encouragement throughout the period this work was carried out. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

I would also like to thank **Dr. R.K Singh** and **Dr.S.K Maity** for his guidance and constructive criticism.

I am grateful to **Prof. K.C. Biswal**, Head of the Department, Chemical Engineering for providing us the necessary opportunities for the completion of our project. I also thank the other staff members of my department for their invaluable help and guidance.

Paramvir Ahlawat (Roll No. 10500012)
B.Tech. Final Yr. Chem. Engg.

CONTENTS

LIST OF FIGURES

ABSTRACT

Computational fluid dynamics (CFD) can predict what will happen , quantitatively , when fluids flow , often with the complication of , simultaneous flow of heat , mass transfer (eg perspiration , dissolution) , phase change (eg melting , freezing , boiling) , chemical reaction (eg combustion , rusting) , mechanical movement , stresses in and displacement of immersed or surroundings solids . Knowing how fluids will flow , and what will be their quantitative effects on the solids with which they are in contact , assists engineers to maximize the yields from their reactors and processing equipment at least cost ; risk.

CFD uses a computer to solve the relevant science-based mathematical equations , using information about the circumstances in question . Its components are therefore : the human being who states the problem , scientific knowledge expressed mathematically , the computer code which embodies this knowledge and expresses the stated problem in scientific terms , the computer hardware which performs the calculations dictated by the software.

The objective of this project is to simulate a gas-solid fluidized by applying CFD techniques in order to investigate hydrodynamics and heat transfer phenomena. Reactor model predictions will be compared with the corresponding experimental data reported in the literature to validate the model . To simulate a gas-solid fluidized bed we need to use the multiphase flow approach . First we have to write the equations for the different flow regimes and then different CFD techniques are applied for discretization of those equations. After that a code is written for calculating the values of volume fraction , velocity and temperature .

CHAPTER 1

LITURATURE REVIEW

1.1 INTRODUCTION

Fluidization is the operation by which particle are transformed into a fluid like state through suspension in a gas or liquid.

Fluidized bed reactors are widely used in the industries due to their superior heat-and mass-transfer as a result of relatively larger particle-fluid contacting area compared to other types of reactors. Therefore fluidized beds are suitable for catalytic reactions especially for exothermic reaction.

Computational fluid dynamic (CFD) is an important tool for design and optimization of chemical processes. A fundamental problem encountered in modeling hydrodynamics of a gas–solid fluidized bed is the motion of two phases . For the fluidization operation minimum two phases are required; one is called primary phase and other one is secondary phase. Usually the fluid which passes through the inlet is considered as primary phase and particulate in the bed is the secondary phase. The operating conditions like superficial fluid velocity, temperature of the primary and secondary phase, and inlet and exit pressure of the bed affect the performance of the fluidized bed. The physical property of the phases, particle size and distribution controls the hydrodynamical behaviors of it. Hydrodynamic modeling has the remarkable ability to synthesize data from various, relatively simple experiments and, thereby, to describe the time-dependent distribution of fluid and solids volume fractions, velocities, pressure, temperatures, and species mass fractions in industrial reactors, where measurement of such quantities might be all but impossible. Such calculations, therefore, allow the designer to visualize the conditions in the reactor, to understand how performance values change as operating conditions are varied, to conduct what-if experiments, and, thereby, to assist in the design process.

Today fluidized bed reactors are used to produce gasoline and other fuels, along with many other chemicals. Many industrially produced polymers are made using fluidized bed reactor technology, such as production of rubber, vinyl chloride, polyethylene, and styrene. Fluidized bed reactors are also used for coal gasification, nuclear power plants, and waste water treatment.

1.2 COMUTATIONAL FLUID DYNAMICS (CFD)

CFD is one of the branches of fluid mechanics that uses numerical methods and algorithms to solve and analyze problems that involve fluid flows. Computers are used to perform the millions of calculations required to simulate the interaction of fluids and gases with the complex surfaces used in engineering. However, even with simplified equations and high speed supercomputers, only approximate solutions can be achieved in many cases. More accurate codes that can accurately and quickly simulate even complex scenarios such as supersonic or turbulent flows are an ongoing area of research.

The fundamental basis of any CFD problem is the Navier-Stokes equations, which define any single-phase fluid flow. These equations can be simplified by removing terms describing viscosity to yield the Euler equations. Further simplification, by removing terms describing vorticity yields the Full Potential equations. Finally, these equations can be linearized to yield the Linearized Potential equations.

The basic procedure for solving any problem in the CFD is as follows :-

1. The geometry of the problem is defined .
2. The volume occupied by the fluid is divided into discrete cells .
3. The mathematical and physical modeling of the problem is defined .
4. Boundary conditions are defined .This involves specifying the fluid behaviour and properties at the boundary of the problem . For transient problems the initial conditions are also applied.
5. The equations are solved iteratively as a steady state or transient .
6. Analysis and visualization of the resulting solution .

1.3 ADVANTAGES OF CFD

Major advancements in the area of gas-solid multiphase flow modeling offer substantial process improvements that have the potential to significantly improve process plant operations. Prediction of gas solid flow fields, in processes such as pneumatic transport lines, risers, fluidized bed reactors, hoppers and precipitators are crucial to the operation of most process plants. Up to now, the inability to accurately model these interactions has limited the role that simulation could play in improving operations. In recent years, computational fluid dynamics (CFD) software developers have focused on this area to develop new modeling methods that can simulate gas-liquid-solid flows to a much higher level of reliability. As a result, process industry engineers are beginning to utilize these methods to make major improvements by evaluating alternatives that would be, if not impossible, too expensive or time-consuming to trial on the plant floor. Over the past few decades, CFD has been used to improve process design by allowing engineers to simulate the performance of alternative configurations, eliminating guesswork that would normally be used to establish equipment geometry and process conditions. The use of CFD enables engineers to obtain solutions for problems with complex geometry and boundary conditions. A CFD analysis yields values for pressure, fluid velocity, temperature, species or phase concentration on a computational grid throughout the solution domain.

The key advantages of the CFD are :-

1. It provides the flexibility to change design parameters without the expense of hardware changes. Hence it costs less than laboratory or field experiments, allowing engineers to try more alternative designs than would be feasible otherwise.

2. It has a faster turnaround time than experiments.

3. It guides the engineer to the root of problems, and is therefore well suited for trouble-shooting .

4. It provides comprehensive information about a flow field, especially in regions where measurements are either difficult or impossible to obtain .

1.4 OBJECTIVE

The objective of this project is to simulate a gas-solid fluidized by applying CFD techniques in order to investigate hydrodynamics phenomena.

1.5 ORGANISATION OF THE REPORT

The present study is about the CFD modeling of the hydrodynamics and heat transfer in fluidized bed reactor . Chapter 2 deals with the multiphase modeling of the fluidized bed reactor which includes the detailed derivation of the continuity and momentum equations . Chapter 3 describes the programming code for the simulation of the fluidized bed reactor . Chapter 4 deals with the results and discussion .

CHAPTER 2

MULTIPHASE MODELING

2.1 MULTIPHASE MODELING APPROACH

Advances in computational fluid mechanics have provided the basis for further insight into the dynamics of the multiphase flow. Currently there are two approaches for the numerical calculations of multiphase flows:

1. Euler – Langrange approach .
2. Euler – Euler approach .

2.1.1 Euler – Langrange Approach :

The Lagrangian discrete phase model follows the Euler-Lagrange approach. The fluid phase is treated as a continuum by solving the time-averaged Navier-Stokes equations, while the dispersed phase is solved by tracking a large number of particles, bubbles, or droplets through the calculated flow field. The dispersed phase can exchange momentum, mass and energy with the fluid phase.

A fundamental assumption made in this model is that the dispersed second phase occupies a low volume fraction, even though high mass loading , $m_{particle} >= m_{fluid}$ is acceptable. The particle or droplet trajectories are computed individually at specified intervals during the fluid phase calculation. This makes the model appropriate for the modeling of spray dryers , coal and liquid fuel combustion , and some particleladen flows, but inappropriate for the modeling of liquid-liquid mixtures, fluidized beds or any application where the volume fraction of the second phase is not negligible.

2.1.2 Euler – Euler Approach :

In the Euler-Euler approach the different phases are treated mathematically as interpenetrating continua. Since the volume of a phase can not be carried occupied by the other phases , the concept of the volume fraction is introduced. These volume fractions are assumed to be continuous functions of space and time and their sum is equal to one. Conservation equations for each phase are derived to obtain a set of equations , which have similar structure for all phases. These equations are closed by providing constitutive relations that are obtained from empirical information or in the case of granular flows by application of kinetic theory.

2.2 GOVERNING EQUATIONS

The basic equation of motion for single phase flows were initially developed by Navier in 1822 in the form of well known Navier – Stokes equation . The Navier – Stokes equations were extended to multiphase flow by using volume averaging of the single phase equation and adding appropriate terms (Drew 1983) . Soo (1967) suggested that particles of identical diameter and density should form a continuum or a particulate phase . It follows that if two different particle size is required in the fluidized bed then there should be two particulate phases . These phases can be treated like a pseudo fluids where they are considered to form the interpenetrating continua . This technique is known as Eulerian – Eulerian method . Witt (1997) showed that the Eulerian – Eulerian method is the most promising method for simulating fluidized bed system .

2.2.1 BASIC APPROACH

It is assumed that the system on which balances are made consists of a sufficient number of particles so that the discontinuities can be smoothed out ; therefore , derivatives of various properties exists and are continuous , unless otherwise specified . Thus for a property per unit volume ψ , the Reynolds transport theorem is used . For unit volume V that may change with time t , there is , for a system bounded by a closed surface , the following mathematical identity (Aris , 1962) :-

$$\frac{d}{d\,t^{i}} \iiint_{V(t)} \psi \; dV = \iiint_{V(t)} \left(\frac{\delta\psi}{\delta t} + \nabla \, . \, \psi \, v_{i} \right) dV \qquad (1.1)$$

In Eq. (1.1) the system moves with velocity $\mathbf{v_i}$. Hence , differentiation with respect to time caries the superscript i to emphasize this effect .

2.2.2 DENSITY AND VOLUME FRACTION

The density for a continuum at a point is defined as :-

$$\rho = \lim_{\partial V \to 0} \frac{\delta m}{\delta V}$$

Now here we are dealing with the raw material so we cannot take the limit to the zero . Now let us define the enough large volume ∂V_0 so that if the volume was increased or decreased slightly it would remain unchanged . So now density can be written as :-

$$\rho = \lim_{\partial V \to \partial V_0} \frac{\delta m}{\delta V}$$

Now let us take a volume ∂V and the volume fraction of the dispersed phase will be :-

$$\varepsilon_N = \lim_{\partial V \to \partial V_0} \frac{\delta V_N}{\delta V}$$

∂V_N = Volume of the dispersed phase . So here the bulk density of the dispersed phase will be

$$\overline{\rho_N} = \lim_{\partial V \to \partial V_0} \frac{\delta m_N}{\delta V}$$

$\delta m_N =$ mass of the dispersed phase particles . $\delta m_N = \rho_N \ \delta V_N , \rho_N =$ actual density

So

$$\overline{\rho_N} = \lim_{\partial V \to \partial V_0} \frac{\rho_N \delta V_N}{\delta V} = \rho_N \lim_{\partial V \to \partial V_0} \frac{\delta V_N}{\delta V} = \rho_N \varepsilon_N$$

Here N represents the phase .

2.2.3 MASS BALANCE

The mass of the fluid can be given in terms of its ρ_i in a multiphase system of volume V as

$$m_i = \iiint_{v(t)} \rho_i \varepsilon_i \, dV \qquad (1.2)$$

The Langrangian mass balance on mass m_i moving with the velocity v_i is

$$\frac{dm_i}{dt} = \frac{d}{dt}\iiint_{v(t)} \rho_i \varepsilon_i \, dV = \iiint m_i' dV \qquad (1.3)$$

where Eq.(1.5) defines the volumetric source of mass m_i' . An application of the Reynolds transport theorem and the usual contradiction argument applied to an arbitrary element of volume gives the well – known continuity equation for the phase i .

$$\frac{\partial(\rho_i \varepsilon_i)}{\partial t} + \nabla.\left(\rho_i \varepsilon_i v_i\right) = m_i' \qquad (1.4)$$

Now according to the conditions since there is no reaction is occurring in $\frac{\partial y}{\partial x}$ the fluidized bed reactor so that the volume source generation term m_i' will not be there . Therefore , the continuity equation can be represented as

$$\frac{\partial(\rho_i \varepsilon_i)}{\partial t} + \nabla.\left(\rho_i \varepsilon_i v_i\right) = 0 \qquad (1.5)$$

For the incompressible fluid ie. for constant density the equation can be represented as ,

$$\frac{\partial(\varepsilon_i)}{\partial t} + \nabla.\left(\varepsilon_i v_i\right) = 0 \qquad (1.6)$$

2.2.4 MOMENTUM BALANCE

The derivation of the momentum balance follows the approach of Bowen (1976) for a multicomponent mixture . The rate of change of momentum of the “particle” or system moving with the velocity v_i equals the sum of the forces acting on the system. In single phase flow , similar to the multicomponent flow ,there exists the force of interaction of phase i with the other phases . In rational mechanics this force is known as the momentum supply . Mathematically this statement can be written as follows . The momentum balance for the phase i is

Rate of change of momentum of phase i = force acting on the phase i

$$\frac{d}{dt}\iiint_{V(t)} \rho_i \varepsilon_i v_i \, dV = p_i \tag{1.7}$$

$p_i =$ external forces(gravitational forces) + surface forces + forces of interaction between the phases

$$p_i = \iiint_{V(t)} \rho_i \varepsilon_i g \, dV + \oiint_{S(t)} \sigma_i da + \iiint f_{ij} dV \tag{1.8}$$

Now an application of the divergence theorem

$$\oiint_{S(t)} \sigma_i da = \iiint_{V(t)} \nabla . \sigma_i . dV \tag{1.9}$$

Hence by the application of Reynolds Transport theorem to the equation 1.7 we get the required momentum equation

$$\frac{\partial(\rho_i \varepsilon_i v_i)}{\partial t} + \nabla . \left(\rho_i \varepsilon_i v_i v_i\right) = \nabla . \sigma_i + \rho_i \varepsilon_i g + f_{ij} \tag{2.0}$$

Here σ_i represents the stress tensor which is given by the expression

$$\sigma_i = -P_i I + \tau_i$$

Where P_i is the pressure and viscous stress tensor , τ_i, is assumed to be of Newtonian form

$$\tau_i = 2\varepsilon_i \mu_i \overline{D} + \varepsilon_i \lambda_i \, tr\left(\overline{D}\right) I$$

Where I is the identity tensor and $\overline{D}$ is the strain tensor for the respective phase , given by

$$\overline{D} = \frac{1}{2}\left[\nabla v_i + (\nabla v_i)^T\right]$$

f_{ij} is an interaction force representing the momentum transfer between gas and solid phase . In this project only drag and buoyancy forces are taken into account . Therefore the interaction between the gas and solid phase is expressed as :

$$f_{ij} = f_{gs} = -\varepsilon_s \nabla P_g - F_{gs}\,(v_s - v_g)$$

Usually F_{gs} is determined using experimental data. Two type of experimental data can be used to calculate the drag coefficient. In the first case when the solid volume fraction is high , the Ergun equation is applied . In the second case a correlation for terminal velocity in the settling beds is expressed as a function of gas volume fraction and Reynolds number . In the second case Syamlal – O'Brien drag model is calculated using terminal velocity as follow

$$F_{gs} = \frac{3\varepsilon_s \varepsilon_g \rho_g}{4{v_r}^2 d_p} CD_s\left(\frac{Re}{v_r}\right)|v_s - v_g|$$

$$v_r = 0.5(A - 0.06Re + \sqrt{(0.06Re)^2 + 0.12Re(2B - A) + A^2}$$

Where ,

$$A = {\varepsilon_g}^{4.14}$$

$$B = \begin{cases} 0.8{\varepsilon_g}^{1.28}\ if\ \varepsilon_g \leq 0.85 \\ {\varepsilon_g}^{4.14} if\ \varepsilon_g > 0.85 \end{cases}$$

$$Re = \frac{d_p|v_s - v_g|\rho_g}{\mu_g}$$

$$CD_s(Re) = \left(0.6 + {}^{4.8}\!/_{\sqrt{Re}}\right)^2$$

Another model for calculating drag force is Gidaspow model

$$F_{gs} = \frac{3\varepsilon_s\varepsilon_g\rho_g}{4d_p} CD_s|v_s - v_g|\varepsilon_g{}^{-2.65} \text{ for } \varepsilon_g > 0.8$$

$$F_{gs} = 150\frac{\varepsilon_s{}^2\mu_g}{\varepsilon_g d_p{}^2} + 1.75\,\frac{\varepsilon_s\rho_g}{d_p}|v_s - v_g| \text{ for } \varepsilon_g \leq 0.8$$

In this project work Gidaspow model is used .

2.2.5 ENERGY BALANCE

The internal energy balance for the gas phase can be written in terms of the gas temperature as follows

$$\varepsilon_g\rho_g C_{pg}\left(\frac{\partial}{\partial t}\,T_g + v_g.\nabla T_g\right) = -H_{sg} - \Delta H_{rg} \qquad (2.1)$$

The solid heat conductivity includes direct conduction through the fractional contact area and indirect conduction through a wedge of the gas that is trapped between the particles . Since the gas heat conductivity is negligible , the heat diffusion term has been ignored . The termal energy for the solid phase is given by

$$\varepsilon_s\rho_s C_{ps}\left(\frac{\partial}{\partial t}\,T_s + v_s.\nabla T_s\right) = \nabla.\varepsilon_s k_s\nabla T_s + H_s - \Delta H_{rs} \qquad (2.2)$$

A number of simplifying assumptions, none of which should be significant in typical applications to fluid-solids reactors, have been made in the formulation of thermal energy equations:-

1) The irreversible rate of increase of internal energy due to viscous dissipation has been neglected. Such terms are negligible except in the case of velocities approaching the speed of sound.

2) The reversible rate of fluid internal energy change due to compression or expansion has been neglected. Such terms will be important in transient, compressible flows.

3) Interfacial flow work terms have not been included, which may lead to a violation of the second law (Lyczkowski, Gidaspow, and Solbrig 1982; Arnold, Drew, and Lahey 1990). This does not necessarily imply large errors in the calculations, because such terms in usual

applications are negligible. Furthermore, a satisfactory formulation including such terms does not exist.

4) The heat of reaction term includes both the enthalpy change due to reaction and the energy transfer because the products and reactants may be at different temperatures.

5) Heat transfer between different solids phases is negligible.

6) Radiative heat transfer is not considered.

.

The heat transfer between the gas and solids is a function of temperature difference between the gas and solid phases .

$$H_{sg} = -\gamma_{gs}^{0}(T_s - T_g)$$

The heat transfer coefficient is related to the particle Nusselt number using the following equation

$$\gamma_{gs}^{0} = \frac{6k_g \varepsilon_s Nu_s}{d_{ps}^2}$$

The Nusselt number is typically determined from one of the many correlations reported in the literature for calculating the heat transfer between particles and fluid in packed or fluidized beds (e.g., Zabrodsky 1966; Gelperin and Einstein 1971; Gunn 1978).

$$Nu_s = \left(7 - 10\varepsilon_g + 5{\varepsilon_g}^2\right)\left(1 + 0.7Re_s^{0.2}Pr^{1/3}\right) + (1.33 - 2.4\varepsilon_g + 102{\varepsilon_g}^2)Re_s^{0.7}Pr^{1/3}$$

Pr is the Prandtl number which is given by

$$Pr = \frac{C_{pg}\mu_g}{k_g}$$

H_{rg} and H_{rs} represents the heat of reaction since there is no reaction is happening so that the both terms are neglected.

CHAPTER 3

NUMERICAL MODELING

3.1 DISCRETIZATION

In mathematics, discretization concerns the process of transferring continuous models and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on digital computers. In order to be processed on a digital computer another process named quantization is essential. The stability of the chosen discretization is generally established numerically rather than analytically as with simple linear problems. Special care must also be taken to ensure that the discretization handles discontinuous solutions gracefully. The Euler equations and Navier-Stokes equations both admit shocks, and contact surfaces.

Some of the discretization methods being used are:

3.1.1 ***Finite Volume Method(FVM)***

This is the "classical" or standard approach used most often in commercial software and research codes. The governing equations are solved on discrete control volumes. FVM recasts the PDE's (Partial Differential Equations) of the N-S equation in the conservative form and then discretize this equation. "Finite volume" refers to the small volume surrounding each node point on a mesh. In the finite volume method, volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem. These terms are then evaluated as fluxes at the surfaces of each finite volume. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative. Another advantage of the finite volume method is that it is easily formulated to allow for unstructured meshes.This guarantees the conservation of fluxes through a particular control volume. Though the overall solution will be conservative in nature there is no guarantee that it is the actual solution. Moreover this method is sensitive to distorted elements which can prevent convergence if such elements are in critical flow regions. This integration approach yields a method that is inherently conservative (i.e. quantities such as density remain physically meaningful)

$$\frac{\partial}{\partial t}\iiint QdV+\iint FdA=0$$

Where Q is the vector of conserved variables, F is the vector of fluxes ,V is the cell volume, and A is the cell surface area.

3.1.2 *Finite Element Method(FEM)*

This method is popular for structural analysis of solids, but is also applicable to fluids. The FEM formulation requires, however, special care to ensure a conservative solution. The FEM formulation has been adapted for use with the Navier-Stokes equations. Although in FEM conservation has to be taken care of, it is much more stable than the FVM approach. Subsequently it is the new direction in which CFD is moving. Generally stability/robustness of the solution is better in FEM though for some cases it might take more memory than FVM methods

In this method, a weighted residual equation is formed:

$$R_i = \iiint W_i Q dV^e$$

where R_i is the equation residual at an element vertex i , Q is the conservation equation expressed on an element basis, W_i is the weight factor and V^e is the volume of the element.

3.1.3 *Finite Difference Method*

This method has historical importance and is simple to program. It is currently only used in few specialized codes. Modern finite difference codes make use of an embedded boundary for handling complex geometries making these codes highly efficient and accurate. Other ways to handle geometries are using overlapping-grids, where the solution is interpolated across each grid. Finite-difference methods approximate the solutions to differential equations by replacing derivative expressions with approximately equivalent difference quotients.

$$f'(a) = \lim_{h\to 0} \frac{f(a+h) - f(a)}{h}$$

then a reasonable approximation for that derivative would be to take

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

for some small value of h. In fact, this is the forward difference equation for the first derivative. Using this and similar formulae to replace derivative expressions in differential equations, one can approximate their solutions without the need for calculus.

3.1.3.1 *Explicit Method*

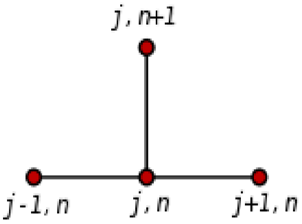
Using a forward difference at time t_n and a second-order central difference for the space derivative at position x_j ("FTCS") we get the recurrence equation:

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}$$

This is an explicit method for solving the one-dimensional heat equation .$\frac{\partial u}{\partial t} = K\frac{\partial^2 u}{\partial x^2}$

We can obtain u_j^{n+1} from the other values this way:

$$u_j^{n+1} = ru_{j+1}^n + (1 - 2r)u_j^n + ru_{j-1}^n$$

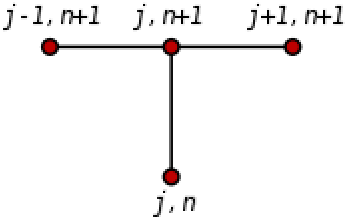


where $r = k / h^2$. So, knowing the values at time n you can obtain the corresponding ones at time n+1 using this recurrence relation.

3.1.3.2 *Implicit Method*

If we use the backward difference at time t_{n+1} and a second-order central difference for the space derivative at position x_j ("BTCS") we get the recurrence equation:

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2}$$



This is an implicit method for solving the one-dimensional heat equation. We can obtain u_j^n from solving a system of linear equations:

$$u_j^n = (1 + 2r)u_j^{n+1} - ru_{j-1}^{n+1} - u_{j+1}^{n+1}$$

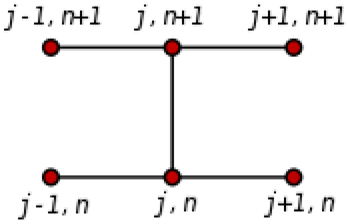
The scheme is always numerically stable and convergent but usually more numerically intensive than the explicit method as it requires solving a system of numerical equations on each time step. The errors are linear over the time step and quadratic over the space step.

3.1.3.2 ***Crank – Nicolson Method***

Finally if we use the central difference at time $t_{n+1/2}$ and a second-order central difference for the space derivative at position x_j ("CTCS") we get the recurrence equation:

$$\frac{u_j^{n+1}-u_j^n}{k}=\frac{1}{2}\left(\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}+\frac{u_{j+1}^{n}-2u_j^{n}+u_{j-1}^{n}}{h^2}\right)$$

This formula is known as the Crank-Nicolson method.



We can obtain u_j^{n+1} from solving a system of linear equations:

$$(2+2r)u_j^{n+1}-ru_{j-1}^{n+1}-u_{j+1}^{n+1}=ru_{j+1}^{n}+(2-2r)u_j^{n}+ru_{j-1}^{n}$$

The scheme is always numerically stable and convergent but usually more numerically intensive as it requires solving a system of numerical equations on each time step. Usually the Crank-Nicolson scheme is the most accurate scheme for small time steps. The explicit scheme is the least accurate and can be unstable, but is also the easiest to implement and the least numerically intensive. The implicit scheme works the best for large time steps.

3.2 DISCRETIZATION OF EQUATIONS

Continuity and momentum equations are discretized with the help of finite difference method since it is easy to operate and also it does not involve any kind of calculus . Explicit method is used for discretization of continuity and momentum equation. For discretization of continuity and momentum equations we need to write the equations in terms of cylindrical coordinates . After that equations will be discretized with the help of central difference technique. At special conditions like at r = 0 where the terms will tend to infinity special techniques are applied like L’Hospital rule is applied at such conditions.

3.2.1 CONTINIUTY EQUATION

Continuity equation for the respective phases can be written as:-

$$\frac{\partial(\rho_i \varepsilon_i)}{\partial t} + \nabla.\left(\rho_i \varepsilon_i v_i\right) = 0$$

So now as we know that in the two phase modeling gaseous and solid phases are present . Here we need to discritize both phase continuity equation so that modeling of each phase should be done. Since we are dealing with the cylindrical bed therefore continuity equation should be written in the form of cylindrical coordinates , which is as follows :-

$$\frac{\partial(\varepsilon_g \rho_g\ )}{\partial t} = -[\ \frac{1}{r}\ \frac{\partial(\varepsilon_g \rho_g\ V_{rg})}{\partial x} + \frac{1}{r}\frac{\partial(\varepsilon_g \rho_g\ V_{\theta g})}{\partial \theta} + \frac{\partial(\ \varepsilon_g \rho_g V_{zg})}{\partial z}\ ]$$

$$\frac{\partial(\varepsilon_g \rho_g\ )}{\partial t} = -[\frac{1}{r}\ \frac{\partial(\varepsilon_g \rho_g\ V_{rg})}{\partial r} + \ \frac{\partial(\ \varepsilon_g \rho_g V_{zg})}{\partial z}\ ]$$

Terms including angular component are to be neglected ,

$$\frac{\partial(\varepsilon_g \rho_g\ )}{\partial t} = -[\frac{1}{r}\ (\ \varepsilon_g\ \rho_g\ \mathrm{r}\ \frac{\partial V_{rg}}{\partial r} + \rho_g\ r\ \frac{\partial \varepsilon_g}{\partial r}\ V_{rg} + \ \varepsilon_g \rho_g V_{rg}\ ) + \varepsilon_g \rho_g \frac{\partial V_{zg}}{\partial z} + \ V_{zg} \rho_g\ \frac{\partial \varepsilon_g}{\partial z}]$$

$$\frac{\partial(\varepsilon_g \rho_g)}{\partial t} = \ -[\ \varepsilon_g \rho_g \frac{\partial V_{rg}}{\partial r} + \ \rho_g V_{rg}\ \frac{\partial \varepsilon_g}{\partial r} + \frac{\varepsilon_g \rho_g V_{rg}}{r} + \rho_g \varepsilon_g \frac{\partial V_{zg}}{\partial z} + \ V_{zg} \rho_g \frac{\partial \varepsilon_g}{\partial z}]$$

Now this equation is discretized by applying explicit method and also with the central difference technique . Discretized equation is written on the next page .

$$\frac{(\varepsilon_g\rho_g)_{i,j}^{t+\Delta t} - (\varepsilon_g\rho_g)_{i,j}^{t}}{\Delta t} =$$

$$[(\varepsilon_g\rho_g)_{i,j}^{t} \frac{\left(V_{rg}{}^{t}{}_{(i+1,j)} - V_{rg}{}^{t}{}_{(i-1,j)}\right)}{2\Delta r} + (V_{rg}\rho_g)^{t}{}_{(i,j)} \frac{\left(\varepsilon_g{}^{t}{}_{(i+1,j)} - \varepsilon_g{}^{t}{}_{(i-1,j)}\right)}{2\Delta r} +$$

$$(\frac{\varepsilon_g\rho_g V_{rg}{}^{t}}{r})_{(i,j)} + (\varepsilon_g\rho_g)^{t}{}_{(i,j)} \frac{\left(V_{zg}{}^{t}{}_{(i,j+1)} - V_{zg}{}^{t}{}_{(i,j-1)}\right)}{2\Delta z} +$$

$$(V_{zg}\rho_g)^{t}{}_{(i,j)} \frac{\left(\varepsilon_g{}^{t}{}_{(i,j+1)} - \varepsilon_g{}^{t}{}_{(i,j-1)}\right)}{2\Delta z}]$$

Now as we know that terms in the continuity equation having r term as a denominator but at r=0 those terms will become equal to infinity so for solving the equations at r=0 we need to apply L'Hospital rule for the discretization of continuity equation. So at r=0 gas phase continuity will become :-

$$\frac{(\varepsilon_g\rho_g)_{i,j}^{t+\Delta t} - (\varepsilon_g\rho_g)_{i,j}^{t}}{\Delta t} =$$

$$[2(\varepsilon_g\rho_g)_{i,j}^{t} \frac{\left(V_{rg}{}^{t}{}_{(i+1,j)} - V_{rg}{}^{t}{}_{(i-1,j)}\right)}{2\Delta r} + (V_{rg}\rho_g)^{t}{}_{(i,j)} \frac{\left(\varepsilon_g{}^{t}{}_{(i+1,j)} - \varepsilon_g{}^{t}{}_{(i-1,j)}\right)}{2\Delta r} +$$

$$((\varepsilon_g\rho_g)^{t}{}_{(i,j)} \frac{\left(V_{zg}{}^{t}{}_{(i,j+1)} - V_{zg}{}^{t}{}_{(i,j-1)}\right)}{2\Delta z} + (V_{zg}\rho_g)^{t}{}_{(i,j)} \frac{\left(\varepsilon_g{}^{t}{}_{(i,j+1)} - \varepsilon_g{}^{t}{}_{(i,j-1)}\right)}{2\Delta z}]$$

This is the discretized gas phase continuity equation . Similarly for the solid phase we can write the discretized solid phase continuity equation with the similar procedure . Discretized solid phase continuity equation is written on the next page .

$$\frac{(\varepsilon_s\rho_s)_{i,j}^{t+\Delta t}-(\varepsilon_g\rho_g\,)_{i,j}^{t}}{\Delta t} \quad =$$

$$[(\varepsilon_s\rho_s\,)_{i,j}^{t}\ \frac{\left({V_{rs}}^{t}{}_{(i+1,j)}-{V_{rs}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+\ (V_{rs}\rho_s)^{t}{}_{(i,j)}\,\frac{\left({\varepsilon_s}^{t}{}_{(i+1,j)}-{\varepsilon_s}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+$$

$$(\frac{\varepsilon_s\rho_s{V_{rs}}^{t}}{r})_{(i,j)}+\ (\varepsilon_s\rho_s)^{t}{}_{(i,j)}\ \frac{\left({V_{zs}}^{t}{}_{(i,j+1)}-{V_{zs}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+$$

$$(V_{zs}\rho_s)^{t}{}_{(i,j)}\ \frac{\left({\varepsilon_s}^{t}{}_{(i,j+1)}-{\varepsilon_s}^{t}{}_{(i,j-1)}\right)}{2\Delta z}]$$

At r = 0

$$\frac{(\varepsilon_s\rho_s)_{i,j}^{t+\Delta t}-(\varepsilon_g\rho_g\,)_{i,j}^{t}}{\Delta t} \quad =$$

$$[2\,(\varepsilon_s\rho_s\,)_{i,j}^{t}\ \frac{\left({V_{rs}}^{t}{}_{(i+1,j)}-{V_{rs}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+\ (V_{rs}\rho_s)^{t}{}_{(i,j)}\,\frac{\left({\varepsilon_s}^{t}{}_{(i+1,j)}-{\varepsilon_s}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+$$

$$+\ (\varepsilon_s\rho_s)^{t}{}_{(i,j)}\ \frac{\left({V_{zs}}^{t}{}_{(i,j+1)}-{V_{zs}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+(V_{zs}\rho_s)^{t}{}_{(i,j)}\ \frac{\left({\varepsilon_s}^{t}{}_{(i,j+1)}-{\varepsilon_s}^{t}{}_{(i,j-1)}\right)}{2\Delta z}]$$

3.2.2 MOMENTUM BALANCE EQUATION

Momentum balance equation for the gas – solid phase in the cylindrical coordinates can be written as :-

r – Momentum Equation

$$\varepsilon_g\rho_g\frac{\partial V_{rg}}{\partial t}+\frac{\varepsilon_g\rho_g V_{rg}}{r}\frac{\partial V_{rg}}{\partial r}+\ \varepsilon_g\rho_g V_{zg}\frac{\partial V_{rg}}{\partial z}$$

$$=\ \frac{1}{r}\,\frac{\partial(r\varepsilon_g\,\tau_{rr})}{\partial r}+\frac{1}{r}\,\frac{\partial(\varepsilon_g\,\tau_{\theta r})}{\partial\theta}+\frac{\partial(\varepsilon_g\,\tau_{rz})}{\partial z}-\frac{\varepsilon_g\tau_{\theta\theta}}{r}-\ f_{gs}$$

$$+\ \varepsilon_g\rho_g g$$

$$\tau_{rr} = 2\mu\frac{\partial V_{rg}}{\partial r} - \frac{2}{3}\mu(\nabla.V)\ ,\ \ \tau_{rz} = +\mu\left[\frac{\partial V_{rg}}{\partial z} + \frac{\partial V_{zg}}{\partial r}\right],,$$

$$\tau_{\theta\theta} = 2\mu\left[\frac{1}{r}\frac{\partial V_{\theta g}}{\partial\theta} + \frac{V_r}{r}\right] - \frac{2}{3}\mu(\nabla.V)$$

Now angular component terms are to be neglected. This is equation is discretized by applying explicit scheme , central difference and forward difference methods .

L.H.S

$$=> (\varepsilon_g\rho_g)_{i,j}^t\ \frac{\left(V_{rg}{}^{t+\Delta t}{}_{(i,j)} - V_{rg}{}^{t}{}_{(i,j)}\right)}{\Delta t} + \left(\frac{\varepsilon_g\rho_g V_{rg}{}^t}{r}\right)_{(i,j)}\frac{\left(V_{rg}{}^{t}{}_{(i,j)} - V_{rg}{}^{t}{}_{(i-1,j)}\right)}{\Delta r} +$$

$$(\ \varepsilon_g\rho_g V_{zg})_{i,j}^t\ \frac{\left(V_{rg}{}^{t}{}_{(i,j)} - V_{rg}{}^{t}{}_{(i,j)}\right)}{\Delta z}$$

R.H.S

$$=> \frac{1}{r}\ \frac{\partial\left(2\mu r\varepsilon_g\frac{\partial V_{rg}}{\partial r}\right)}{\partial r} + \frac{\partial\left(\mu\varepsilon_g\left[\frac{\partial V_{rg}}{\partial z} + \frac{\partial V_{zg}}{\partial r}\right]\right)}{\partial r} - \ 2\frac{\mu\varepsilon_g V_{rg}}{r} - \varepsilon_g\nabla P_g$$

$$+F_{gs}(V_s - V_g) + \ \varepsilon_g\rho_g g$$

$$=>\frac{2\mu}{r}\left[\varepsilon_g\frac{\partial V_{rg}}{\partial r} + r\varepsilon_g\frac{\partial^2 V_{rg}}{\partial r^2} + \ r\ \frac{\partial V_{rg}}{\partial r}\ \frac{\partial\varepsilon_g}{\partial r}\right] + \ \mu\ \left[\frac{\partial V_{rg}}{\partial z}\ \frac{\partial\varepsilon_g}{\partial z} + \frac{\partial V_{zg}}{\partial r}\ \frac{\partial\varepsilon_g}{\partial z} + \right.$$

$$\left.\varepsilon_g\frac{\partial^2 V_{rg}}{\partial z^2} + \varepsilon_g\frac{\partial^2 V_{zg}}{\partial z\,\partial r}\ \right] - 2\varepsilon_g\mu\frac{V_{rg}}{r} - \varepsilon_g\frac{\partial P_g}{\partial r} + F_{gs}\left(V_s - V_g\right) + \ \varepsilon_g\rho_g g$$

$$=>\frac{2\mu}{r}\varepsilon_g\frac{\partial V_{rg}}{\partial r} + 2\mu\varepsilon_g\frac{\partial^2 V_{rg}}{\partial r^2} + \ 2\mu\frac{\partial V_{rg}}{\partial r}\frac{\partial\varepsilon_g}{\partial r} + \mu\frac{\partial V_{rg}}{\partial z}\ \frac{\partial\varepsilon_g}{\partial z} + \mu\frac{\partial V_{zg}}{\partial r}\ \frac{\partial\varepsilon_g}{\partial z} +$$

$$\mu\varepsilon_g\frac{\partial^2 V_{rg}}{\partial z^2} + \mu\frac{\partial^2 V_{zg}}{\partial z\,\partial r} - 2\varepsilon_g\mu\frac{V_{rg}}{r} - \varepsilon_g\frac{\partial P_g}{\partial r} + F_{gs}\left(V_s - V_g\right) + \ \varepsilon_g\rho_g g$$

$$-\frac{2}{3r}\frac{\partial}{\partial r}\left\{\mu r\varepsilon_g(\nabla.V)\right\} + \ \ \frac{2\mu}{3r}\varepsilon_g(\nabla.V)$$

Now discretization,

$$(2\mu\,\frac{\varepsilon_g}{r})_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+(2\mu\varepsilon_g\,)_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i-1,j)}\right)}{\nabla r^2}\ +$$

$$(2\mu\,)_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}\frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+$$

$$(\mu\,)_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i,j+1)}-{V_{rg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}\frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)}-{\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+\ (\mu\,)_{i,j}^{t}\ \frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)}-{\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z}$$

$$\frac{\left({V_{zg}}^{t}{}_{(i+1,j)}-{V_{zg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}\ +(\mu\varepsilon_g\,)_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i,j+1)}-{V_{rg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i,j-1)}\right)}{\nabla z^2}+$$

$$(\mu\,\frac{\varepsilon_g}{2\,\nabla r})_{i,j}^{t}\ \frac{\left({V_{zg}}^{t}{}_{(i+1,j+1)}-{V_{zg}}^{t}{}_{(i+1,j-1)}-{V_{zg}}^{t}{}_{(i-1,j+1)}-{V_{zg}}^{t}{}_{(i-1,j-1)}\right)}{2\,\Delta z}-2(\varepsilon_g\mu\frac{V_{rg}}{r})_{i,j}^{t}\ -$$

$$(\varepsilon_g\,)_{i,j}^{t}\ \frac{\left({P_{rg}}^{t}{}_{(i,j+1)}-{P_{rg}}^{t}{}_{(i,j-1)}\right)}{\Delta r}+(\varepsilon_g\rho_g\,g)_{i,j}^{t}\ +\ \{F_{gs}\,(V_{rs}-V_{rg})\}_{i,j}^{t}\ +$$

$$(\frac{2}{3}\mu)_{i,j}^{t}\{\ \frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r}\frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+(\frac{V_{rg}}{r})_{i,j}^{t}\ \frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\,\Delta r}$$

$$+(2\,\frac{\varepsilon_g}{r})_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+\ \ (\varepsilon_g\,)_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i-1,j)}\right)}{\nabla r^2}+$$

$$(\frac{\varepsilon_g}{r})_{i,j}^{t}\ \frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-{V_{zg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+\frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\,\Delta r}\ \frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-{V_{zg}}^{t}{}_{(i,j-1)}\right)}{2\,\Delta z}+$$

$$(\varepsilon_g\,)_{i,j}^{t}\ \frac{\left({V_{zg}}^{t}{}_{(i+\mathbf{1},\boldsymbol{j}+\mathbf{1})}-\ {V_{zg}}^{t}{}_{(i+\mathbf{1},\boldsymbol{j}-\mathbf{1})}-{V_{zg}}^{t}{}_{(i-\mathbf{1},\boldsymbol{j}+\mathbf{1})}-\ {V_{zg}}^{t}{}_{(i-\mathbf{1},\boldsymbol{j}-\mathbf{1})}\right)}{4\Delta r\Delta z}\}+$$

$$\{\left(\frac{V_{rg}}{r^2}\right)_{i,j}^{t}\ +\frac{\mathbf{1}}{r}\frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+\frac{\mathbf{1}}{r}\ \frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-{V_{zg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}\}\ (\frac{2}{3}\,\varepsilon_g\mu)_{i,j}^{t}$$

At r =0 by applying L'Hospital rule ,

L.H.S

$$=> (\varepsilon_g \rho_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t+\Delta t}{}_{(i,j)} - {V_{rg}}^{t}{}_{(i,j)}\right)}{\Delta t} + (\varepsilon_g \rho_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i,j)} - {V_{rg}}^{t}{}_{(i-1,j)}\right)^{2}}{\Delta r} +$$

$$(\varepsilon_g \rho_g V_{zg})_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i,j)} - {V_{rg}}^{t}{}_{(i,j)}\right)}{\Delta z}$$

R.H.S

$$(\mu \frac{\varepsilon_g}{\Delta r})_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+2,j)} - 2{V_{rg}}^{t}{}_{(i,j)} + {V_{rg}}^{t}{}_{(i-2,j)}\right)}{2\Delta r} +$$

$$(2\mu\varepsilon_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - 2{V_{rg}}^{t}{}_{(i,j)} + {V_{rg}}^{t}{}_{(i-1,j)}\right)}{\nabla r^{2}} +$$

$$(2\mu)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - {V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r} \frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)} - {\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r} +$$

$$(\mu)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i,j+1)} - {V_{rg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z} \frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)} - {\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z} + (\mu)_{i,j}^{t} \frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)} - {\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z}$$

$$\frac{\left({V_{zg}}^{t}{}_{(i+1,j)} - {V_{zg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r} + (\mu\varepsilon_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i,j+1)} - {V_{rg}}^{t}{}_{(i,j)} + {V_{rg}}^{t}{}_{(i,j-1)}\right)}{\nabla z^{2}} +$$

$$(\mu \frac{\varepsilon_g}{2\nabla r})_{i,j}^{t} \frac{\left({V_{zg}}^{t}{}_{(i+1,j+1)} - {V_{zg}}^{t}{}_{(i+1,j-1)} - {V_{zg}}^{t}{}_{(i-1,j+1)} - {V_{zg}}^{t}{}_{(i-1,j-1)}\right)}{2\Delta z} - 2(\varepsilon_g \mu \frac{V_{rg}}{r})_{i,j}^{t} -$$

$$(\varepsilon_g)_{i,j}^{t} \frac{\left({P_{rg}}^{t}{}_{(i,j+1)} - {P_{rg}}^{t}{}_{(i,j-1)}\right)}{\Delta r} + (\varepsilon_g \mu)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - {V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r} +$$

$$\{F_{gs} \left(V_{rs} - V_{rg}\right)\}_{i,j}^{t} + (\frac{2}{3}\mu)_{i,j}^{t} \{ \frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)} - {\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r} \frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - {V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r} +$$

$$\frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - {V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r} \frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)} - {\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r} + (\varepsilon_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+2,j)} - {V_{rg}}^{t}{}_{(i,j)} + {V_{rg}}^{t}{}_{(i-2,j)}\right)}{2\nabla r^{2}} +$$

$$(\varepsilon_g)_{i,j}^{t} \frac{\left({V_{rg}}^{t}{}_{(i+1,j)} - {V_{rg}}^{t}{}_{(i,j)} + {V_{rg}}^{t}{}_{(i-1,j)}\right)}{\nabla r^{2}} +$$

$$+\frac{\left({\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}\right)}{2\Delta r}\;\frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-{V_{zg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+$$

$$(\varepsilon_g\,)_{i,j}^{t}\;\frac{\left({V_{zg}}^{t}{}_{(i+\mathbf{1},j+\mathbf{1})}-\;{V_{zg}}^{t}{}_{(i+\mathbf{1},j-\mathbf{1})}-{V_{zg}}^{t}{}_{(i-\mathbf{1},j+\mathbf{1})}-\;{V_{zg}}^{t}{}_{(i-\mathbf{1},j-\mathbf{1})}\right)}{2\Delta r\Delta z}\}+$$

$$\{\frac{\left({V_{rg}}^{t}{}_{(i+2,j)}-{V_{rg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i-2,j)}\right)}{(4\Delta r)(2\Delta r)}+\frac{\left({V_{rg}}^{t}{}_{(i+2,j)}-{V_{rg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i-2,j)}\right)}{(2\Delta r)(2\Delta r)}+$$

$$\frac{\left({V_{zg}}^{t}{}_{(i+1,j+1)}-{V_{zg}}^{t}{}_{(i+1,j-1)}-{V_{zg}}^{t}{}_{(i-1,j+1)}-{V_{zg}}^{t}{}_{(i-1,j-1)}\right)}{2\Delta r.2\Delta z}\}\;\left(\tfrac{2}{3}\,\varepsilon_g\mu\right)_{i,j}^{t}$$

Z – Momentum Equation

$$\varepsilon_g\rho_g\frac{\partial v_{zg}}{\partial t}+\varepsilon_g\rho_g V_{rg}\frac{\partial V_{zg}}{\partial r}+\varepsilon_g\rho_g V_{zg}\frac{\partial V_{zg}}{\partial z}=\;\frac{\mathbf{1}}{r}\;\frac{\partial(\,r\,\varepsilon_g\tau_{rz}\,)}{\partial r}+$$

$$\frac{\mathbf{1}}{r}\;\frac{\partial(\,\tau_{\theta z}\,\varepsilon_g)}{\partial\theta}+\;\frac{\partial(\,\varepsilon_g\tau_{zz})}{\partial z}-\;\varepsilon_g\frac{\partial\rho_g}{\partial z}+F_{gs}\left(V_s-V_g\right)+\varepsilon_g\rho_g g$$

$$\tau_{r_z}=\;\mu\left[\frac{\partial v_{rg}}{\partial z}+\frac{\partial v_{zg}}{\partial r}\right],\;\;\tau_{z_z}=2\mu\;\frac{\partial v_{rg}}{\partial z}-\;\frac{2}{3}\mu(\,\nabla.V)$$

Now angular component terms are to be neglected. This is equation is discretized by applying explicit scheme , central difference and forward difference methods.

L.H.S

$$=(\varepsilon_g\rho_g)_{(i,j)}^{t}\;\frac{({V_{zg}}_{(i,j)}^{(t+\Delta t)}-{V_{zg}}_{(i,j)}^{(t)}\;)}{\Delta t}+(\varepsilon_g V_r\rho_g)_{(i,j)}^{t}\;\frac{({V_{zg}}_{(i,j)}^{(t)}-{V_{zg}}_{(i-\mathbf{1},j)}^{(t)}\;)}{\Delta r}$$

$$+\;(\varepsilon_g V_z\rho_g)_{(i,j)}^{t}\,\frac{({V_{zg}}_{(i,j)}^{(t)}-{V_{zg}}_{(i-\mathbf{1},j)}^{(t)}\;)}{\Delta z}$$

R.H.S

$$= \frac{\mu}{r}\left[\frac{\partial\left(r\,\varepsilon_g \frac{\partial V_{rg}}{\partial z}\right)}{\partial r} + \frac{\partial\left(r\,\varepsilon_g \frac{\partial V_{zg}}{\partial r}\right)}{\partial r} + \frac{\partial\left(2\mu\,\varepsilon_g \frac{\partial V_{zg}}{\partial z}\right)}{\partial r} - \varepsilon_g \frac{\partial \rho_g}{\partial z} + \right.$$

$$\left. F_{gs}\left(V_s - V_g\right) + \varepsilon_g\rho_g g - \frac{\partial(\varepsilon_g \frac{2}{3}\mu(\nabla.V)}{\partial z} \right]$$

$$\Rightarrow \frac{\mu}{r}\ [\left[r\frac{\partial \varepsilon_g}{\partial r}\,\frac{\partial V_{rg}}{\partial z} + \varepsilon_g \frac{\partial V_{rg}}{\partial z} + r\varepsilon_g \frac{\partial^2 V_{rg}}{\partial z \partial r} \right] + \ [\mu\varepsilon_g \frac{\partial^2 V_{rg}}{\partial z \partial r} +$$

$$\frac{\mu}{r}\varepsilon_g \frac{\partial V_{rg}}{\partial r} + \mu\frac{\partial \varepsilon_g}{\partial r}\,\frac{\partial V_{zg}}{\partial r} + \mu\varepsilon_g \frac{\partial^2 V_{rg}}{\partial r^2}] + 2\mu\varepsilon_g \frac{\partial^2 V_{rg}}{\partial r^2} - \varepsilon_g \frac{\partial \rho_g}{\partial z} +$$

$$F_{gs}\left(V_s - V_g\right) + \varepsilon_g\rho_g g - \frac{\partial(\varepsilon_g \frac{2}{3}\mu(\nabla.V)}{\partial z}$$

$$\Rightarrow \mu_{(i,j)}\left(\frac{{\varepsilon_g}^{t}_{(i+1,j)} - {\varepsilon_g}^{t}_{(i-1,j)}}{2\Delta r}\right)\left(\frac{{V_{rg}}^{t}_{(i,j+1)} - {V_{rg}}^{t}_{(i,j-1)}}{2\Delta z}\right) +$$

$$\mu{\varepsilon_g}^{t}_{(i,j)}\left(\frac{{V_{zg}}^{t}_{(i+1,j)} - 2{V_{zg}}^{t}_{(i,j)} + {V_{zg}}^{t}_{(i-1,j)}}{(\Delta r)^2}\right) +$$

$$\frac{\mu{\varepsilon_g}^{t}_{(i,j)}}{2\Delta r}\left[\frac{{V_{rg}}^{t}_{(i+1,j+1)} - {V_{rg}}^{t}_{(i+1,j-1)}}{2\Delta z} - \frac{{V_{rg}}^{t}_{(i-1,j+1)} - {V_{rg}}^{t}_{(i-1,j-1)}}{2\Delta z}\right] +$$

$$\frac{\mu}{r}{\varepsilon_g}^{t}_{(i,j)}\left(\frac{{V_{rg}}^{t}_{(i,j+1)} - {V_{rg}}^{t}_{(i,j-1)}}{2\Delta z}\right) +$$

$$\frac{\mu}{r}{\varepsilon_g}^{t}_{(i,j)}\left(\frac{{V_{zg}}^{t}_{(i+1,j)} - {V_{zg}}^{t}_{(i-1,j)}}{2\Delta r}\right) +$$

$$2\mu{\varepsilon_g}^{t}_{(i,j)}\left(\frac{{V_{zg}}^{t}_{(i,j+1)} - 2{V_{zg}}^{t}_{(i,j)} + {V_{zg}}^{t}_{(i,j-1)}}{(\Delta z)^2}\right) + F_{gs}\left(V_s - V_g\right) +$$

$$\varepsilon_g\rho_g g\,{}^{t}_{(i,j)} + 0.3{\varepsilon_g}^{t}_{(i,j)} -$$

$$(\tfrac{2}{3}\mu)^{t}_{i,j}\{\frac{\left({\varepsilon_g}^{t}_{(i,j+1)} - {\varepsilon_g}^{t}_{(i,j-1)}\right)}{2\Delta z}\frac{\left({V_{rg}}^{t}_{(i+1,j)} - {V_{rg}}^{t}_{(i-1,j)}\right)}{2\Delta r} +$$

$$(\frac{V_{rg}}{r})_{i,j}^{t}\ \frac{\left({\varepsilon_g}^{t}{}_{(\boldsymbol{i},j+1)} - {\varepsilon_g}^{t}{}_{(\boldsymbol{i},j-1)}\right)}{2\Delta z} +$$

$$\frac{\left({\varepsilon_g}^{t}{}_{(\boldsymbol{i},j+1)} - {\varepsilon_g}^{t}{}_{(\boldsymbol{i},j-1)}\right)}{2\Delta z}\frac{\left({V_{zg}}^{t}{}_{(\boldsymbol{i},j+1)} - {V_{zg}}^{t}{}_{(\boldsymbol{i},j-1)}\right)}{2\Delta z} +$$

$$(\varepsilon_g\ )_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(\boldsymbol{i}+1,j+1)} - {V_{rg}}^{t}{}_{(\boldsymbol{i}+1,j-1)} - {V_{rg}}^{t}{}_{(\boldsymbol{i}-1,j+1)} - {V_{rg}}^{t}{}_{(\boldsymbol{i}-1,j-1)}\right)}{2\,\Delta r.2\Delta z} +$$

$$(\frac{\varepsilon_g}{r})_{i,j}^{t}\ \frac{\left({V_{rg}}^{t}{}_{(\boldsymbol{i},j+1)} - {V_{rg}}^{t}{}_{(\boldsymbol{i},j-1)}\right)}{2\Delta z} + \quad (\varepsilon_g\ )_{i,j}^{t}\ \frac{\left({V_{zg}}^{t}{}_{(\boldsymbol{i},j+1)} - 2{V_{zg}}^{t}{}_{(\boldsymbol{i},j)} + {V_{rg}}^{t}{}_{(\boldsymbol{i},j-1)}\right)}{\Delta z^2}$$

At r= 0

L.H.S

$$= (\varepsilon_g\rho_g)_{(i,j)}^{t}\ \frac{({V_{zg}}_{(i,j)}^{(t+\Delta t)} - {V_{zg}}_{(i,j)}^{(t)}\ )}{\Delta t} + (\varepsilon_g V_r \rho_g)_{(i,j)}^{t}\ \frac{({V_{zg}}_{(i,j)}^{(t)} - {V_{zg}}_{(i-\mathbf{1},j)}^{(t)}\ )}{\Delta r}$$

$$+ (\varepsilon_g V_z \rho_g)_{(i,j)}^{t}\frac{({V_{zg}}_{(i,j)}^{(t)} - {V_{zg}}_{(i-\mathbf{1},j)}^{(t)}\ )}{\Delta z}$$

R.H.S

$$= \frac{\mu}{r}\Bigg[\ \frac{\partial\left(\ r\ \varepsilon_g\frac{\partial V_{rg}}{\partial z}\right)}{\partial r} + \frac{\partial\left(\ r\ \varepsilon_g\frac{\partial V_{zg}}{\partial r}\right)}{\partial r} + \frac{\partial\left(\ 2\mu\ \varepsilon_g\frac{\partial V_{zg}}{\partial z}\right)}{\partial r} - \ \varepsilon_g\frac{\partial\rho_g}{\partial z} +$$

$$F_{gs}\left(V_s - V_g\right) + \varepsilon_g\rho_g g - \frac{\partial(\varepsilon_g\frac{2}{3}\mu(\ \nabla.V)}{\partial z}\ \Bigg]$$

$$\Rightarrow \frac{\mu}{r}\ [\Big[\ r\frac{\partial\varepsilon_g}{\partial r}\ \frac{\partial V_{rg}}{\partial z} + \varepsilon_g\frac{\partial V_{rg}}{\partial z} + \ r\varepsilon_g\frac{\partial^2 V_{rg}}{\partial z\partial r}\ \Big] + \ [\mu\varepsilon_g\frac{\partial^2 V_{rg}}{\partial z\partial r} +$$

$$\frac{\mu}{r}\varepsilon_g\frac{\partial V_{rg}}{\partial r} + \mu\frac{\partial\varepsilon_g}{\partial r}\ \frac{\partial V_{zg}}{\partial r} + \ \mu\varepsilon_g\frac{\partial^2 V_{rg}}{\partial r^2}] + 2\mu\varepsilon_g\frac{\partial^2 V_{rg}}{\partial r^2} - \varepsilon_g\frac{\partial\rho_g}{\partial z} +$$

$$F_{gs}\left(V_s - V_g\right) + \varepsilon_g\rho_g g - \frac{\partial(\varepsilon_g\frac{2}{3}\mu(\ \nabla.V)}{\partial z}$$

$$\mu_{(i,j)}\left(\frac{{\varepsilon_g}^{t}{}_{(i+1,j)}-{\varepsilon_g}^{t}{}_{(i-1,j)}}{2\Delta r}\right)\left(\frac{{V_{rg}}^{t}{}_{(i,j+1)}-{V_{rg}}^{t}{}_{(i,j-1)}}{2\Delta z}\right)+$$

$$\mu{\varepsilon_g}^{t}{}_{(i,j)}\left(\frac{{V_{zg}}^{t}{}_{(i+1,j)}\ -2{V_{zg}}^{t}{}_{(i,j)}\ +{V_{zg}}^{t}{}_{(i-1,j)}}{(\Delta r)^2}\right)+$$

$$\frac{\mu{\varepsilon_g}^{t}_{(i,j)}}{2\Delta r}\left[\frac{{V_{rg}}^{t}{}_{(i+1,j+1)}-{V_{rg}}^{t}{}_{(i+1,j-1)}}{2\Delta z}-\frac{{V_{rg}}^{t}{}_{(i-1,j+1)}-{V_{rg}}^{t}{}_{(i-1,j-1)}}{2\Delta z}\right]+$$

$$\mu{\varepsilon_g}^{t}_{(i,j)}\left[\frac{{V_{rg}}^{t}{}_{(i+1,j+1)}-{V_{rg}}^{t}{}_{(i+1,j-1)}-{V_{rg}}^{t}{}_{(i-1,j+1)}-{V_{rg}}^{t}{}_{(i-1,j-1)}}{2\Delta z\,2\Delta r}\right]+$$

$$(\varepsilon_g\mu\,)^{t}_{i,j}\ \frac{\left({V_{zg}}^{t}{}_{(i+2,j)}-2{V_{zg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i-2,j)}\right)}{\Delta r.2\Delta z}$$

$$+2\mu{\varepsilon_g}^{t}{}_{(i,j)}\left(\frac{{V_{zg}}^{t}{}_{(i,j+1)}\ -2{V_{zg}}^{t}{}_{(i,j)}\ +{V_{zg}}^{t}{}_{(i,j-1)}}{(\Delta z)^2}\right)+F_{gs}\left(V_s-V_g\right)+$$

$$\varepsilon_g\rho_g g\ {}^{t}_{(i,j)}+0.3{\varepsilon_g}^{t}_{(i,j)}-$$

$$(\tfrac{2}{3}\,\mu)^{t}_{i,j}\,\{\frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)}-{\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z}\frac{\left({V_{rg}}^{t}{}_{(i+1,j)}-{V_{rg}}^{t}{}_{(i-1,j)}\right)}{2\Delta r}+$$

$$(\varepsilon_g\,)^{t}_{i,j}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j+1)}-{V_{rg}}^{t}{}_{(i+1,j-1)}-{V_{rg}}^{t}{}_{(i-1,j+1)}-{V_{rg}}^{t}{}_{(i-1,j-1)}\right)}{2\Delta r.2\Delta z}+$$

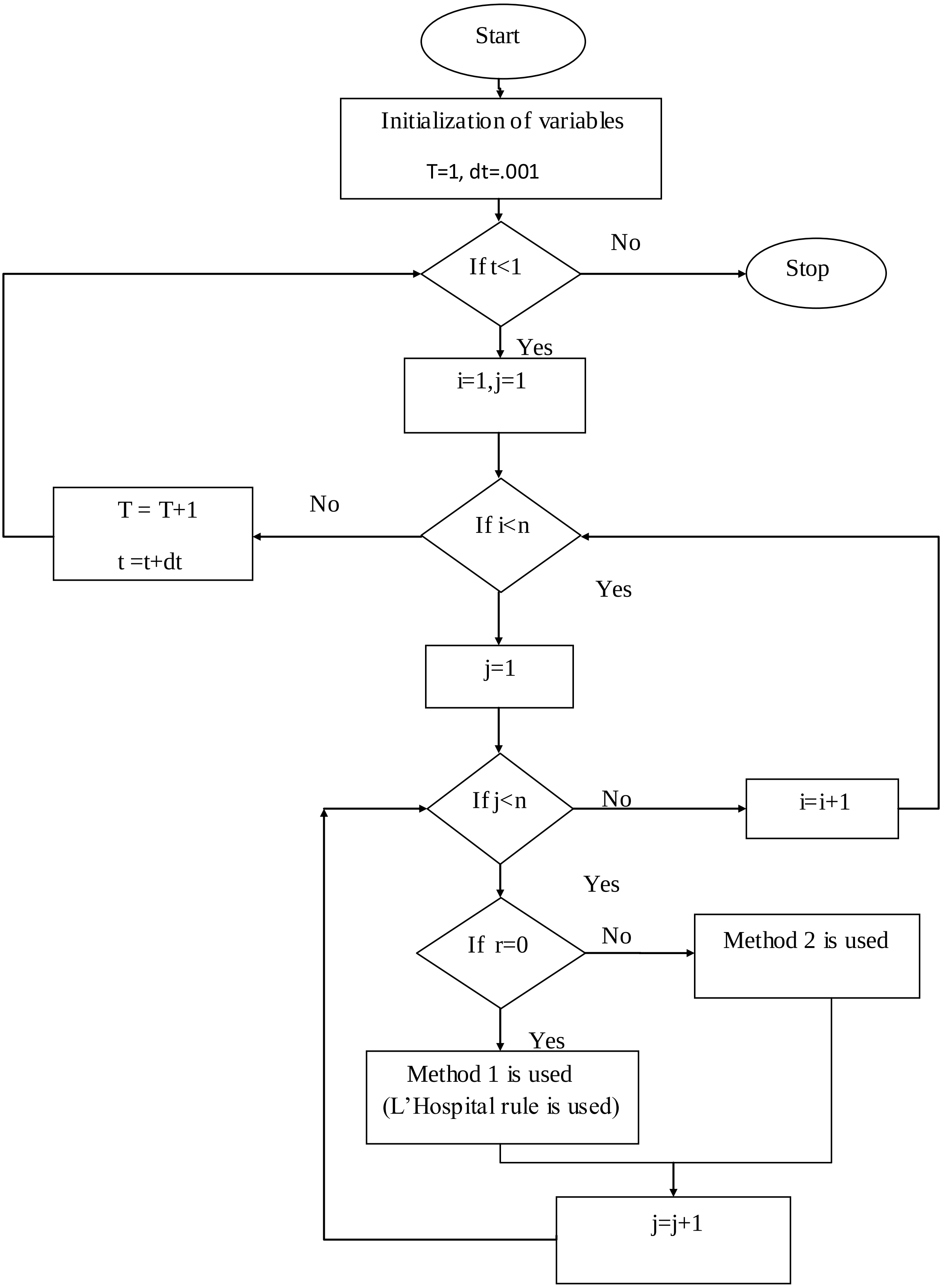
$$\frac{\left({\varepsilon_g}^{t}{}_{(i,j+1)}-{\varepsilon_g}^{t}{}_{(i,j-1)}\right)}{2\Delta z}\frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-{V_{zg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+$$

$$(\varepsilon_g\,)^{t}_{i,j}\ \frac{\left({V_{rg}}^{t}{}_{(i+1,j+1)}-{V_{rg}}^{t}{}_{(i+1,j-1)}-{V_{rg}}^{t}{}_{(i-1,j+1)}-{V_{rg}}^{t}{}_{(i-1,j-1)}\right)}{2\Delta r.2\Delta z}+$$

$$(\frac{\varepsilon_g}{r})^{t}_{i,j}\ \frac{\left({V_{rg}}^{t}{}_{(i,j+1)}-{V_{rg}}^{t}{}_{(i,j-1)}\right)}{2\Delta z}+$$

$$(\varepsilon_g\,)^{t}_{i,j}\ \frac{\left({V_{zg}}^{t}{}_{(i,j+1)}-2{V_{zg}}^{t}{}_{(i,j)}+{V_{rg}}^{t}{}_{(i,j-1)}\right)}{\Delta z^2}$$

3.3 FLOW CHART



3.4 MATLAB CODE

A Matlab code is written for the modeling of discretized equations.

```
 Eg=zeros(50,101,101);Vrg=zeros(50,101,101);Vzg=zeros(50,101,101);
 Vrs=zeros(50,101,101);Vzs=zeros(50,101,101);Es=zeros(50,101,101);
 t=1;dt=.002;row_g=90;row_s=2500;dz=0.1;dr=0.56;mu=0.1;
 g=9.81;dp=0.0275;
 for i=2:100

   for j=2:100

       Es(t,i,j)=0.6;
       Eg(t,i,j)=0.4;
       Vrg(t,i,j)=0;
       Vrs(t,i,j)=0;
       Vzs(t,i,j)=0;
       Vzg(t,i,j)=0;

   end
       Vzg(t,i,1)=0.38;
       Vzg(t+1,i,1)=0.38;
       Vrg(t,i,101)=0;
       Vrs(t,i,101)=0;
       Vzg(t,i,101)=0;
       Vzs(t,i,101)=0;
       Es(t,i,101)=0;
       Eg(t,i,101)=0;

       Vrg(t+1,i,101)=0;
       Vrs(t+1,i,101)=0;
       Vzg(t+1,i,101)=0;
       Vzs(t+1,i,101)=0;
            Es(t+1,i,101)=0;
            Eg(t+1,i,101)=0;
       end

while t<50
   for i=2:100
       Vzg(t,i,1)=0.38;
       Vzg(t+1,i,1)=0.38;
       Vrg(t,i,101)=0;
       Vrs(t,i,101)=0;
       Vzg(t,i,101)=0;
      Vzs(t,i,101)=0;
       Es(t,i,101)=0;
       Eg(t,i,101)=0;

       Vrg(t+1,i,101)=0;
       Vrs(t+1,i,101)=0;
       Vzg(t+1,i,101)=0;
       Vzs(t+1,i,101)=0;
       Es(t+1,i,101)=0;
       Eg(t+1,i,101)=0;
   end

for j=1:101

   Vrg(t,101,j)=0;
   Vrs(t,101,j)=0;
   Vzg(t,101,j)=0;
   Vzs(t,101,j)=0;
   Es(t,101,j)=0;
   Eg(t,101,j)=0;
   Vrg(t,1,j)=0;
   Vrs(t,1,j)=0;
   Vzs(t,1,j)=0;
   Vzg(t,1,j)=0;
   Es(t,1,j)=0;
   Eg(t,1,j)=0;

  Vrg(t+1,101,j)=0;
  Vrs(t+1,101,j)=0;
  Vzg(t+1,101,j)=0;
  Vzs(t+1,101,j)=0;
  Es(t+1,101,j)=0;
  Eg(t+1,101,j)=0;
  Vrg(t+1,1,j)=0;
  Vrs(t+1,1,j)=0;
  Vzs(t+1,1,j)=0;
  Es(t+1,1,j)=0;
  Eg(t+1,1,j)=0;
end


for i=51:100
```

```
 for j=2:100


         x1=Eg(t,i,j)*row_g*( (Vrg(t,i+1,j)-Vrg(t,i-1,j)) /(2*dr) );

         x2=row_g*Vrg(t,i,j)*( (Eg(t,i+1,j)-Eg(t,i-1,j)) /(2*dr) );

         y1=Es(t,i,j)*row_s*( (Vrs(t,i+1,j)-Vrs(t,i-1,j)) /(2*dr) );

         y2=row_s*Vrs(t,i,j)*( (Es(t,i+1,j)-Es(t,i-1,j)) /(2*dr) );



         x3=row_g*Eg(t,i,j)*( (Vzg(t,i,j+1)-Vzg(t,i,j-1)) /(2*dz) );

         x4=row_g*Vzg(t,i,j)*( (Eg(t,i,j+1)-Eg(t,i,j-1)) /(2*dz) );

         y3=row_s*Es(t,i,j)*( (Vzs(t,i,j+1)-Vzs(t,i,j-1)) /(2*dz) );

         y4=row_s*Vzs(t,i,j)*( (Es(t,i,j+1)-Es(t,i,j-1)) /(2*dz) );


if i==51
  x5=Eg(t,i,j)*row_g*( (Vrg(t,i+1,j)-Vrg(t,i-1,j)) /(2*dr) );
  y5=Es(t,i,j)*row_s*( (Vrs(t,i+1,j)-Vrs(t,i-1,j)) /(2*dr) );

else

    x5=(Eg(t,i,j)*Vrg(t,i,j)*row_g)/((i-51)*dr);
    y5=(Es(t,i,j)*Vrs(t,i,j)*row_s)/((i-51)*dr);
end
     Eg(t+1,i,j)=( ( ( -1*( x1+x2+x3+x4+x5 ) )*dt )+( Eg(t,i,j)*row_g ) ) /row_g;

     Es(t+1,i,j)=( ( ( -1*( y1+y2+y3+y4+y5 ) )*dt )+( Es(t,i,j)*row_s ) ) /row_s;

     if Eg(t+1,i,j)<0||Eg(t+1,i,j)>1
       Eg(t+1,i,j)=0;
     end
     if Es(t+1,i,j)<0||Es(t+1,i,j)>1
       Es(t+1,i,j)=0;
     end
    end
end
%///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////%

for i=51:100

    for j=2:100
    if i==51
        y9=(Es(t,i,j)*row_s*((Vrs(t,i,j)-Vrs(t,i-1,j))*(Vrs(t,i,j)-Vrs(t,i-1,j)))))/dr*dr;
        x1=mu*Es(t,i,j)*(Vrs(t,i+2,j)-2*Vrs(t,i,j)+Vrs(t,i-2,j))/2*dr*dr;
        y5=Es(t,i,j)*(Vrs(t,i+2,j)-2*Vrs(t,i,j)+Vrs(t,i-2,j))/2*dr*dr*2;
        y6=Es(t,i,j)*(Vzs(t,i+1,j+1)-Vzs(t,i-1,j+1)-Vzs(t,i+1,j-1)+Vzs(t,i-1,j-1))/2*2*dz*dr;
        x9=(Vrs(t,i+1,j)-Vrs(t,i-1,j))*(Es(t,i+1,j)-Es(t,i-1,j))/2*2*dr*dr;
        x10=2*Es(t,i,j)*(Vrs(t,i+2,j)-2*Vrs(t,i,j)+Vrs(t,i-2,j))/2*dr*dr*2;
        y2=Es(t,i,j)*(Vzs(t,i+1,j+1)-Vzs(t,i-1,j+1)-Vzs(t,i+1,j-1)+Vzs(t,i-1,j-1))/2*2*dz*dr;
        y0=Es(t,i,j)*(Vrs(t,i+2,j)-2*Vrs(t,i,j)+Vrs(t,i-2,j))/8*dr*dr;
        x0=Es(t,i,j)*mu*(Vrs(t,i+1,j)-Vrs(t,i-1,j))/dr;
    else

        x1=((mu*Es(t,i,j))*(Vrs(t,i+1,j)-Vrs(t,i-1,j)))/((i-51)*dr*dr);
        y9=(Es(t,i,j)*row_s*Vrs(t,i,j)*(Vrs(t,i,j)-Vrs(t,i-1,j)))/((i-51)*dr*dr);
        y5=(Es(t,i,j)*(Vrs(t,i+1,j)-Vrs(t,i-1,j)))/(2*dr*(i-51)*dr);
        x9=(Vrs(t,i,j)*(Es(t,i+1,j)-Es(t,i-1,j)))/(2*dr*(i-51)*dr);
        x10=(Es(t,i,j)*(Vrs(t,i+1,j)-Vrs(t,i-1,j)))/(dr*(i-51)*dr);
        y2=(Es(t,i,j)*(Vzs(t,i,j+1)-Vzs(t,i,j-1)))/(2*dz*(i-51)*dr);
        y6=(Es(t,i,j)*(Vzs(t,i,j+1)-Vzs(t,i,j-1)))/(2*dz*(i-51)*dr);
        y0=Es(t,i,j)*Vrs(t,i,j)/(i-51)*dr*(i-51)*dr;
        x0=2*Es(t,i,j)*mu*Vrs(t,i,j)/(i-51)*dr;
    end
        x2=((2*mu*Es(t,i,j))*(Vrs(t,i+1,j)-(2*Vrs(t,i,j))+Vrs(t,i-1,j)))/(dr*dr);
        x3=(mu*(Vrs(t,i+1,j)-Vrs(t,i-1,j))*(Es(t,i+1,j)-Es(t,i-1,j)))/(2*dr*dr);
        x8=((Es(t,i+1,j)-Es(t,i-1,j))*(Vrs(t,i+1,j)-Vrs(t,i-1,j)))/(4*dr*dr);
        y1=(Es(t,i,j)*(Vrs(t,i+1,j)-(2*Vrs(t,i,j))+Vrs(t,i-1,j)))/(dr*dr);
        x12=(Es(t,i,j)*row_s*g);


        temp=Vrs(t,i,j)-Vrg(t,i,j);
        if temp<0
        temp=0-temp;
        end
```

```
x11=(((151*Es(t,i,j)*Es(t,i,j)*mu)/(Eg(t,i,j)*dp*dp))+((1.75*Es(t,i,j)*row_g*temp)/dp))*(Vr
s(t,i,j)-Vrg(t,i,j));


            x4=(mu*(Es(t,i,j+1)-Es(t,i,j-1))*(Vrs(t,i,j+1)-Vrs(t,i,j-1)))/(4*dz*dz);
            x5=(mu*(Es(t,i,j+1)-Es(t,i,j-1))*(Vzs(t,i+1,j)-Vzs(t,i-1,j)))/(4*dr*dz);
            x6=(mu*Es(t,i,j)*(Vrs(t,i,j+1)-(2*Vrs(t,i,j))+Vrs(t,i,j-1)))/(dz*dz);
            x7=(mu*Es(t,i,j)*(Vzs(t,i+1,j+1)-Vzs(t,i+1,j-1)-Vzs(t,i-1,j+1)+Vzs(t,i-1,j-1) )
)/(4*dr*dz);
            y3=((Es(t,i+1,j)-Es(t,i-1,j))*(Vzs(t,i,j+1)-Vzs(t,i,j-1)))/(4*dz*dr);
            y4=(Es(t,i,j)*(Vzs(t,i+1,j+1)-Vzs(t,i+1,j-1)-Vzs(t,i-1,j+1)+Vzs(t,i-1,j-1)))/(4*dz*dr);

            y11=(Vzs(t,i,j)*Es(t,i,j)*row_s*(Vrs(t,i,j)-Vrs(t,i,j-1)))/(dz);




%     /*RHS2*/
y7=(2*mu*(-x8-x9-x10-y1-y2-y3-y4+y5+y6+y0)/3);

%     /*RHS1*/
     y8=x1+x2+x3+x4+x5+x6+x7-x11+x12-x0;

Vrs(t+1,i,j)=(((y7+y8-y9-y11)*dt)/(Es(t,i,j)*row_s))+Vrs(t,i,j);

if Vrs(t+1,i,j)<0
    Vrs(t+1,i,j)=0;
end
    end
end
%
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////%

for i=51:100

    for j=2:100
    if i==51
        y9=(Eg(t,i,j)*row_g*((Vrg(t,i,j)-Vrg(t,i-1,j))*(Vrg(t,i,j)-Vrg(t,i-1,j)))))/dr*dr;
        x1=mu*Eg(t,i,j)*(Vrg(t,i+2,j)-2*Vrg(t,i,j)+Vrg(t,i-2,j))/2*dr*dr;
        y5=Eg(t,i,j)*(Vrg(t,i+2,j)-2*Vrg(t,i,j)+Vrg(t,i-2,j))/2*dr*dr*2;
        y6=Eg(t,i,j)*(Vzg(t,i+1,j+1)-Vzg(t,i-1,j+1)-Vzg(t,i+1,j-1)+Vzg(t,i-1,j-1))/2*2*dz*dr;
        x9=(Vrg(t,i+1,j)-Vrg(t,i-1,j))*(Eg(t,i+1,j)-Eg(t,i-1,j))/2*2*dr*dr;
        x10=2*Eg(t,i,j)*(Vrg(t,i+2,j)-2*Vrg(t,i,j)+Vrg(t,i-2,j))/2*dr*dr*2;
        y2=Eg(t,i,j)*(Vzg(t,i+1,j+1)-Vzg(t,i-1,j+1)-Vzg(t,i+1,j-1)+Vzg(t,i-1,j-1))/2*2*dz*dr;
        y0=Eg(t,i,j)*(Vrg(t,i+2,j)-2*Vrg(t,i,j)+Vrg(t,i-2,j))/8*dr*dr;
        x0=Eg(t,i,j)*mu*(Vrg(t,i+1,j)-Vrg(t,i-1,j))/dr;
    else

        x1=((mu*Eg(t,i,j))*(Vrg(t,i+1,j)-Vrg(t,i-1,j)))/((i-51)*dr*dr);
        y9=(Eg(t,i,j)*row_g*Vrg(t,i,j)*(Vrg(t,i,j)-Vrg(t,i-1,j)))/((i-51)*dr*dr);
        y5=(Eg(t,i,j)*(Vrg(t,i+1,j)-Vrg(t,i-1,j)))/(2*dr*(i-51)*dr);
        x9=(Vrg(t,i,j)*(Eg(t,i+1,j)-Eg(t,i-1,j)))/(2*dr*(i-51)*dr);
        x10=(Eg(t,i,j)*(Vrg(t,i+1,j)-Vrg(t,i-1,j)))/(dr*(i-51)*dr);
        y2=(Eg(t,i,j)*(Vzg(t,i,j+1)-Vzg(t,i,j-1)))/(2*dz*(i-51)*dr);
        y6=(Eg(t,i,j)*(Vzg(t,i,j+1)-Vzg(t,i,j-1)))/(2*dz*(i-51)*dr);
        y0=Eg(t,i,j)*Vrg(t,i,j)/(i-1)*dr*(i-51)*dr;
        x0=2*Eg(t,i,j)*mu*Vrg(t,i,j)/(i-51)*dr;
    end
        x2=((2*mu*Eg(t,i,j))*(Vrg(t,i+1,j)-(2*Vrg(t,i,j))+Vrg(t,i-1,j)))/(dr*dr);
        x3=(mu*(Vrg(t,i+1,j)-Vrg(t,i-1,j))*(Eg(t,i+1,j)-Eg(t,i-1,j)))/(2*dr*dr);
        x8=((Eg(t,i+1,j)-Eg(t,i-1,j))*(Vrg(t,i+1,j)-Vrg(t,i-1,j)))/(4*dr*dr);
        y1=(Eg(t,i,j)*(Vrg(t,i+1,j)-(2*Vrg(t,i,j))+Vrg(t,i-1,j)))/(dr*dr);
        x12=(Eg(t,i,j)*row_g*g);


        temp=Vrs(t,i,j)-Vrg(t,i,j);
        if temp<0
        temp=0-temp;
        end

x11=(((151*Es(t,i,j)*Es(t,i,j)*mu)/(Eg(t,i,j)*dp*dp))+((1.75*Es(t,i,j)*row_g*temp)/dp))*(Vr
s(t,i,j)-Vrg(t,i,j));

            x4=(mu*(Eg(t,i,j+1)-Eg(t,i,j-1))*(Vrg(t,i,j+1)-Vrg(t,i,j-1)))/(4*dz*dz);
            x5=(mu*(Eg(t,i,j+1)-Eg(t,i,j-1))*(Vzg(t,i+1,j)-Vzg(t,i-1,j)))/(4*dr*dz);
            x6=(mu*Eg(t,i,j)*(Vrg(t,i,j+1)-(2*Vrg(t,i,j))+Vrg(t,i,j-1)))/(dz*dz);
            x7=(mu*Eg(t,i,j)*(Vzg(t,i+1,j+1)-Vzg(t,i+1,j-1)-Vzg(t,i-1,j+1)+Vzg(t,i-1,j-1) )
)/(4*dr*dz);
            y3=((Eg(t,i+1,j)-Eg(t,i-1,j))*(Vzg(t,i,j+1)-Vzg(t,i,j-1)))/(4*dz*dr);
            y4=(Eg(t,i,j)*(Vzg(t,i+1,j+1)-Vzg(t,i+1,j-1)-Vzg(t,i-1,j+1)+Vzg(t,i-1,j-1)))/(4*dz*dr);

            y11=(Vzg(t,i,j)*Eg(t,i,j)*row_g*(Vrg(t,i,j)-Vrg(t,i,j-1)))/(dz);
```

```
%      /*RHS2*/
y7=(2*mu*(-x8-x9-x10-y1-y2-y3-y4+y5+y6+y0)/3);

%      /*RHS1*/
    y8=x1+x2+x3+x4+x5+x6+x7+x11+x12-x0;


Vrg(t+1,i,j)=(((y7+y8-y9-y11)*dt)/(Eg(t,i,j)*row_g))+Vrg(t,i,j);
 if Vrg(t+1,i,j)<0
    Vrg(t+1,i,j)=0;
 end


    end
end

%//////////////////////////////////////////////////////////////////////////
%////////////////////////////////////////////////////////////////////
for i=51:100

    for j=2:100

        if i==51
            x2=mu*Eg(t,i,j)*(Vrg(t,i+1,j+1)-Vrg(t,i-1,j+1)-Vrg(t,i+1,j-1)+Vrg(t,i-1,j-1))/4*dr*dz;
            x4=mu*Eg(t,i,j)*(Vzg(t,i+2,j)-2*Vzg(t,i,j)+Vzg(t,i-2,j))/4*dr*dr;
            y2=(Vrg(t,i+1,j)-Vrg(t,i-1,j))*(Eg(t,i,j+1)-Eg(t,i,j-1))/4*dr*dz;
            y5=Eg(t,i,j)*(Vrg(t,i+1,j+1)-Vrg(t,i-1,j+1)-Vrg(t,i+1,j-1)+Vrg(t,i-1,j-1))/4*dr*dz;
        else

        y5=(Eg(t,i,j)*(Vrg(t,i,j+1)-Vrg(t,i,j-1)))/(2*(i-51)*dr*dz);
        y2=(Vrg(t,i,j)*(Eg(t,i,j+1)-Eg(t,i,j-1)))/(2*dz*(i-51)*dr);
        x2=(mu*Eg(t,i,j)*(Vrg(t,i,j+1)-Vrg(t,i,j-1)))/(2*(i-51)*dr*dz);
        x4=(mu*Eg(t,i,j)*(Vzg(t,i+1,j)-Vzg(t,i-1,j)))/(2*(i-51)*dr*dr);

        end

        x5=(mu*(Eg(t,i+1,j)-Eg(t,i-1,j))*(Vzg(t,i+1,j)-Vzg(t,i-1,j)))/(4*dr*dr);
        x6=(mu*Eg(t,i,j)*(Vzg(t,i+1,j)-(2*Vzg(t,i,j))+Vzg(t,i-1,j)))/(dr*dr);

        x8=(Eg(t,i,j)*0.3);
        temp=Vzs(t,i,j)-Vzg(t,i,j);
        if temp<0
```

```
  temp=0-temp;
      end


x9=(((151*Es(t,i,j)*Es(t,i,j)*mu)/(Eg(t,i,j)*dp*dp))+((1.75*Es(t,i,j)*row_g*temp)/dp))*(Vzs(
t,i,j)-Vzg(t,i,j));



        x1=(mu*(Eg(t,i+1,j)-Eg(t,i-1,j))*(Vrg(t,i,j+1)-Vrg(t,i,j-1)))/(4*dr*dz);
        x3=(mu*Eg(t,i,j)*(Vrg(t,i+1,j+1)-Vrg(t,i+1,j-1)-Vrg(t,i-1,j+1)+Vrg(t,i-1,j-1)
))/(4*dr*dz);
        x7=(2*mu*Eg(t,i,j)*(Vzg(t,i,j+1)-(2*Vzg(t,i,j))+Vzg(t,i,j-1)))/(dz*dz);
        y1=(Eg(t,i,j+1)-Eg(t,i,j-1))*(Vrg(t,i+1,j)-Vrg(t,i-1,j))/(4*dr*dz);
        y3=(Eg(t,i,j+1)-Eg(t,i,j-1))*(Vzg(t,i,j+1)-Vzg(t,i,j-1))/(4*dz*dz);
        y4=(Eg(t,i,j)*(Vrg(t,i+1,j+1)-Vrg(t,i+1,j-1)-Vrg(t,i-1,j+1)+Vrg(t,i-1,j-1)))/(4*dr*dz);
        y6=(Eg(t,i,j)*(Vzg(t,i,j+1)-(2*Vzg(t,i,j))+Vzg(t,i,j-1)))/(dz*dz);
% /*LHS2*/
y9=(Eg(t,i,j)*row_g*Vzg(t,i,j)*(Vzg(t,i,j)-Vzg(t,i,j-1)))/dz;

%   /*RHS2*/
y7=(2*mu*(y1+y2+y3+y4+y5+y6))/3;

%   /*RHS1*/
x11=x1+x2+x3+x4+x5+x6+x7+x8+(Eg(t,i,j)*row_g*g)+x9;

%  /*LHS1*/
y8=(Eg(t,i,j)*Vrg(t,i,j)*row_g*(Vzg(t,i,j)-Vzg(t,i-1,j)))/dr;

     Vzg(t+1,i,j)=(((x11-y7-y8-y9)*dt)/(Eg(t,i,j)*row_g))+Vzg(t,i,j);
     if Vzg(t+1,i,j)<0
        Vzg(t+1,i,j)=0;
     end
   end

end
%
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
for i=51:100

   for j=2:100
```

```
    if i==51
      x2=mu*Es(t,i,j)*(Vrs(t,i+1,j+1)-Vrs(t,i-1,j+1)-Vrs(t,i+1,j-1)+Vrs(t,i-1,j-1))/4*dr*dz;
      x4=mu*Es(t,i,j)*(Vzs(t,i+2,j)-2*Vzs(t,i,j)+Vzs(t,i-2,j))/4*dr*dr;
      y2=(Vrs(t,i+1,j)-Vrs(t,i-1,j))*(Es(t,i,j+1)-Es(t,i,j-1))/4*dr*dz;
      y5=Es(t,i,j)*(Vrs(t,i+1,j+1)-Vrs(t,i-1,j+1)-Vrs(t,i+1,j-1)+Vrs(t,i-1,j-1))/4*dr*dz;
    else

    y5=(Es(t,i,j)*(Vrs(t,i,j+1)-Vrs(t,i,j-1)))/(2*(i-51)*dr*dz);
    y2=(Vrs(t,i,j)*(Es(t,i,j+1)-Es(t,i,j-1)))/(2*dz*(i-51)*dr);
    x2=(mu*Es(t,i,j)*(Vrs(t,i,j+1)-Vrs(t,i,j-1)))/(2*(i-51)*dr*dz);
    x4=(mu*Es(t,i,j)*(Vzs(t,i+1,j)-Vzs(t,i-1,j)))/(2*(i-51)*dr*dr);

    end

    x5=(mu*(Es(t,i+1,j)-Es(t,i-1,j))*(Vzs(t,i+1,j)-Vzs(t,i-1,j)))/(4*dr*dr);
    x6=(mu*Es(t,i,j)*(Vzs(t,i+1,j)-(2*Vzs(t,i,j))+Vzs(t,i-1,j)))/(dr*dr);

    x8=(0.6-Eg(t,i,j)*0.3);
    temp=Vzs(t,i,j)-Vzg(t,i,j);
    if temp<0
    temp=0-temp;
    end


x9=(((151*Es(t,i,j)*Es(t,i,j)*mu)/(Eg(t,i,j)*dp*dp))+((1.75*Es(t,i,j)*row_g*temp)/dp))*(Vzs(
t,i,j)-Vzg(t,i,j));




      x1=(mu*(Es(t,i+1,j)-Es(t,i-1,j))*(Vrs(t,i,j+1)-Vrs(t,i,j-1)))/(4*dr*dz);
      x3=(mu*Es(t,i,j)*(Vrs(t,i+1,j+1)-Vrs(t,i+1,j-1)-Vrs(t,i-1,j+1)+Vrs(t,i-1,j-1)
))/(4*dr*dz);
      x7=(2*mu*Es(t,i,j)*(Vzs(t,i,j+1)-(2*Vzs(t,i,j))+Vzs(t,i,j-1)))/(dz*dz);
      y1=(Es(t,i,j+1)-Es(t,i,j-1))*(Vrs(t,i+1,j)-Vrs(t,i-1,j))/(4*dr*dz);
      y3=(Es(t,i,j+1)-Es(t,i,j-1))*(Vzs(t,i,j+1)-Vzs(t,i,j-1))/(4*dz*dz);
      y4=(Es(t,i,j)*(Vrs(t,i+1,j+1)-Vrs(t,i+1,j-1)-Vrs(t,i-1,j+1)+Vrs(t,i-1,j-1)))/(4*dr*dz);
      y6=(Es(t,i,j)*(Vzs(t,i,j+1)-(2*Vzs(t,i,j))+Vzs(t,i,j-1)))/(dz*dz);
% /*LHS2*/
y9=(Es(t,i,j)*row_s*Vzs(t,i,j)*(Vzs(t,i,j)-Vzs(t,i,j-1)))/dz;

%   /*RHS2*/
y7=(2*mu*(y1+y2+y3+y4+y5+y6))/3;
```

```
%   /*RHS1*/
x11=x1+x2+x3+x4+x5+x6+x7+x8+(Es(t,i,j)*row_s*g)+x9;

%  /*LHS1*/
y8=(Es(t,i,j)*Vrs(t,i,j)*row_s*(Vzs(t,i,j)-Vzs(t,i-1,j)))/dr;

        Vzs(t+1,i,j)=(((x11-y7-y8-y9)*dt)/(Es(t,i,j)*row_s))+Vzs(t,i,j);
        if Vzs(t+1,i,j)<0
            Vzs(t+1,i,j)=0;
        end
    end

end
n=1;
while n<50
  for j=2:100

    Eg(t+1,51-n,j)=Eg(t+1,51+n,j);
    Es(t+1,51-n,j)=Es(t+1,51+n,j);
    Vrg(t+1,51-n,j)=Vrg(t+1,51+n,j);
    Vzg(t+1,51-n,j)=Vzg(t+1,51+n,j);
    Vrs(t+1,51-n,j)=Vrs(t+1,51+n,j);
    Vzs(t+1,51-n,j)=Vzs(t+1,51+n,j);


  end
   n=n+1;
end
t=t+1;
end
```

This code is written for calculating the values of volume fractions of solid, fluid phase, velocity of solid and fluid phase in the radial and vertical direction.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULTS

This section includes the results obtained by simulation of the code. This part involves how the properties are changing with radial distance , vertical direction , time and how they are varying from point to point at a specific time. Properties like fluid volume fraction , solid volume fraction , velocity of the fluid phase , velocity of the solid phase are plotted with time , radial distance and vertical direction.

4.1.1 Fluid volume fraction

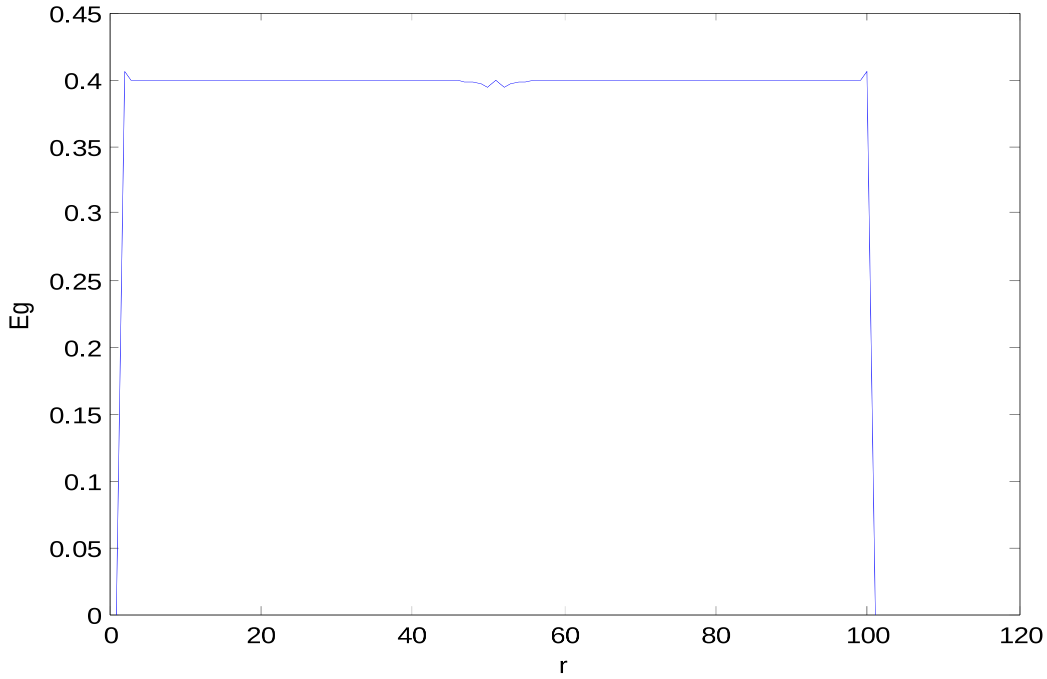


Fig 1.1 PLOT OF FLUID VOLUME FRACTION vs RADIAL DISTANCE

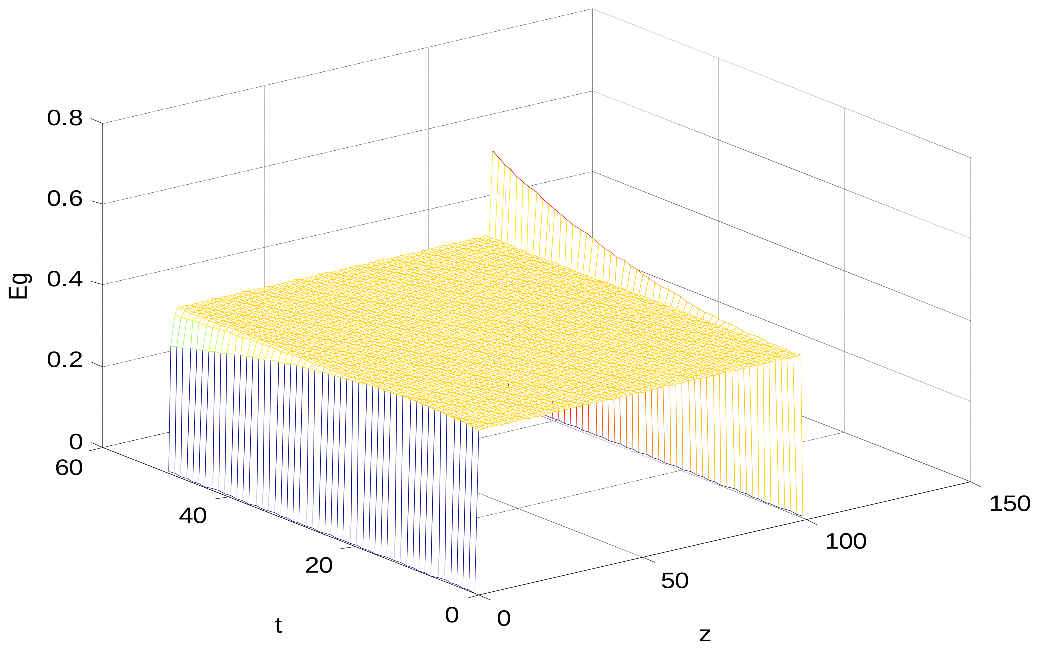


Fig 1.2 FLUID VOLUME FRACTION vs HEIGHT vs TIME

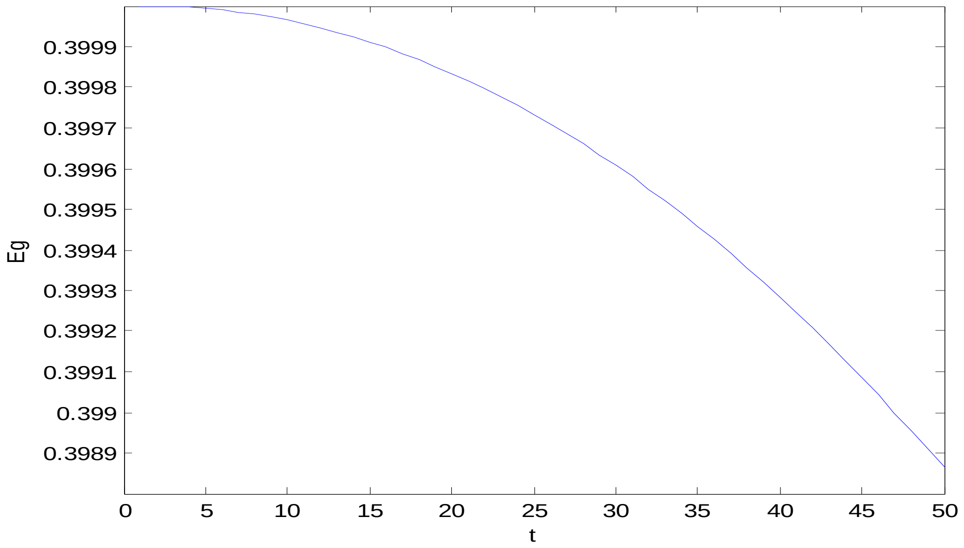


Fig 1.3 FLUID VOLUME FRACTION vs TIME

4.1.2 Solid volume fraction

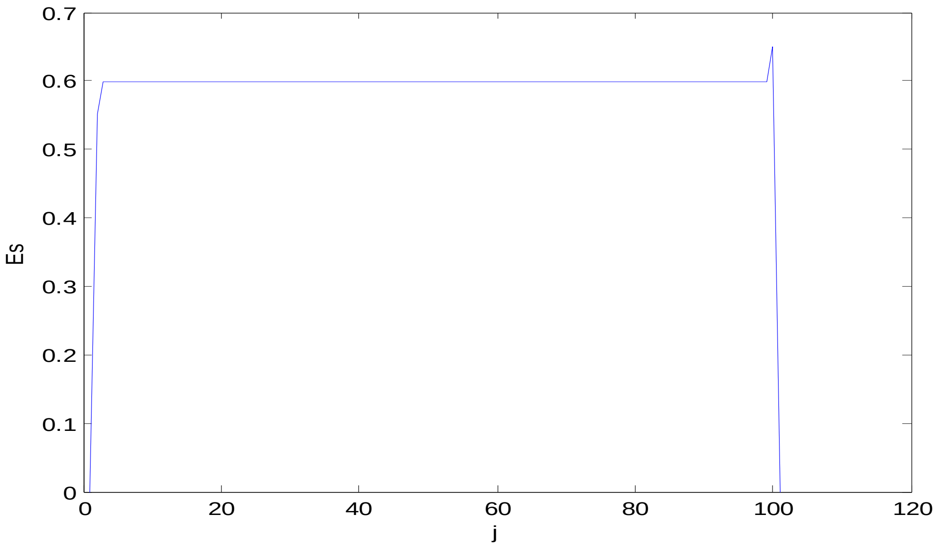


Fig 1.4 PLOT OF SOLID VOLUME FRACTION vs HEIGHT

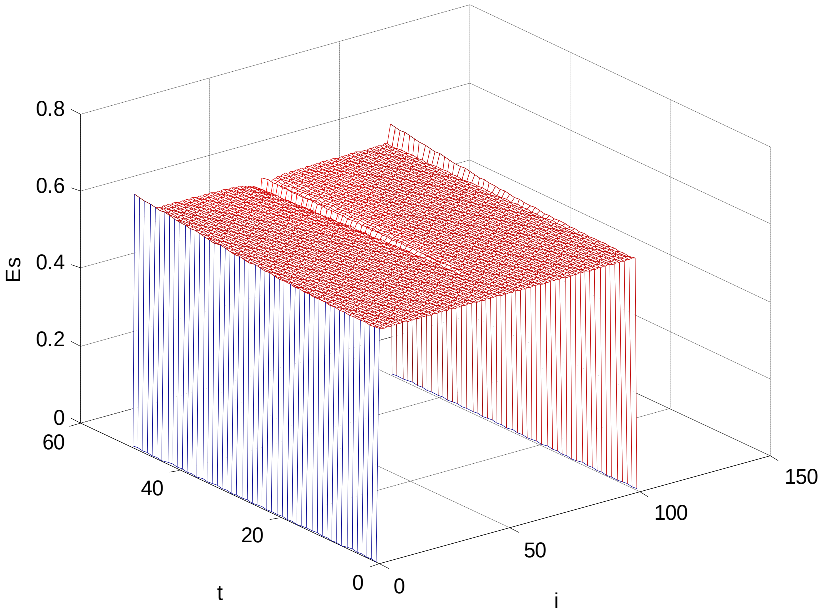


Fig 1.5 PLOT OF SOLID VOLUME FRACTION vs TIME vs RAIDAL DIRECTION

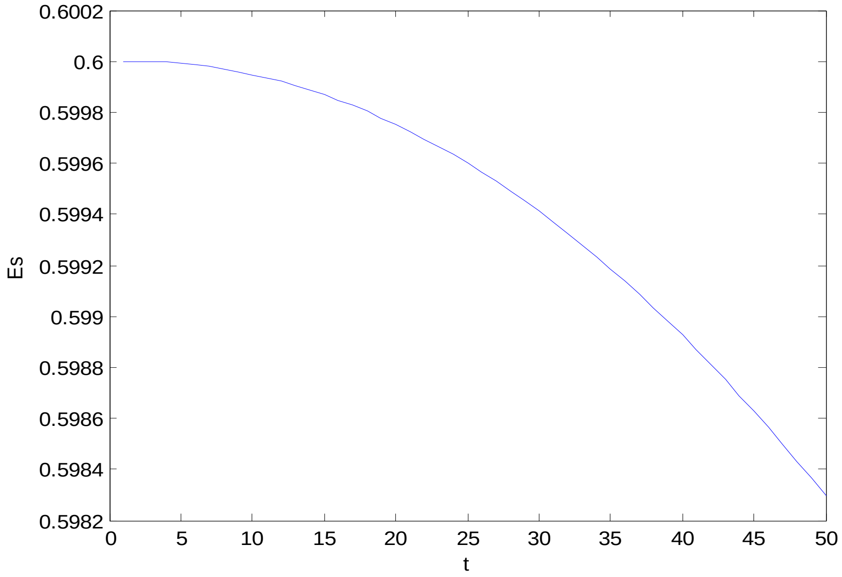


Fig 1.6 PLOT OF SOLID VOLUME FRACTION vs TIME

4.1.3 Fluid phase velocity

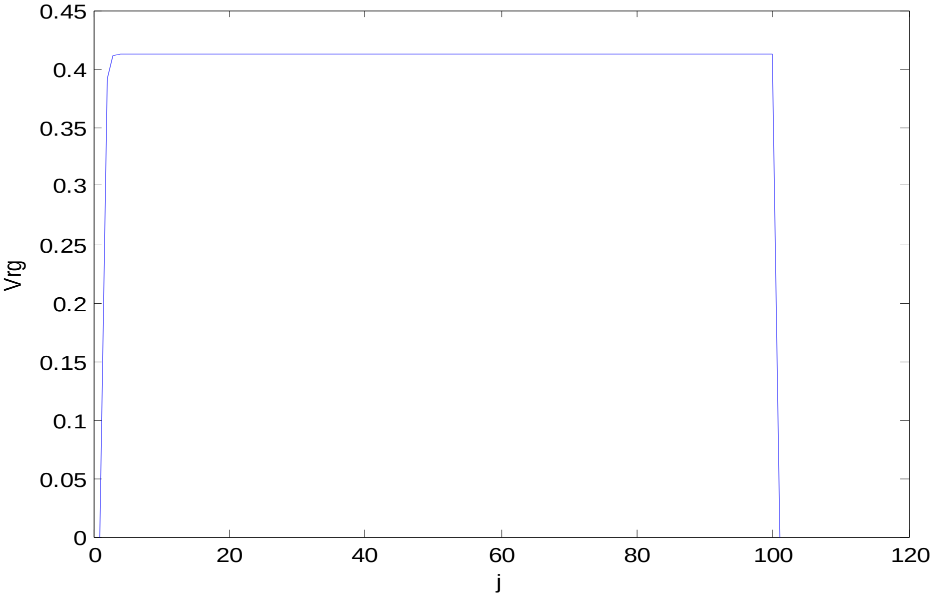


Fig 1.7 FLUID PHASE RADIAL VELOCITY vs HEIGHT

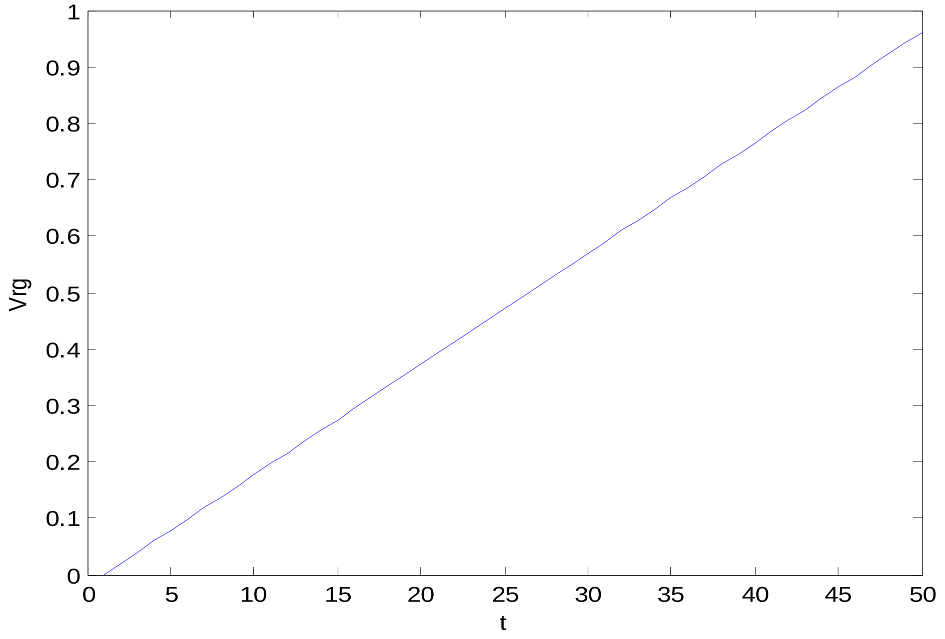


Fig 1.8 FLUID PHASE RADIAL VELOCITY vs TIME

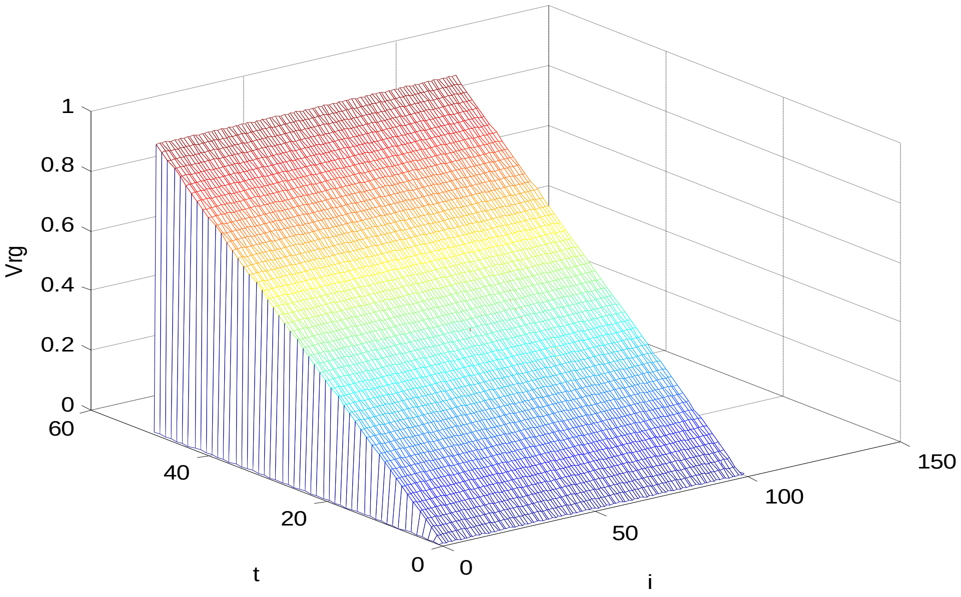


Fig 1.9 FLUID PHASE RADIAL VELOCITY vs TIME vs RADIAL DISTANCE

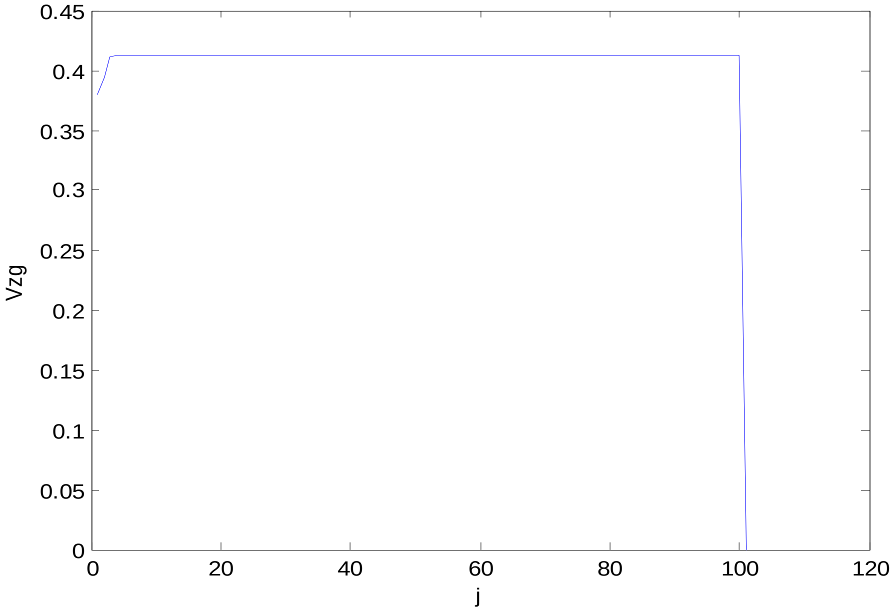


Fig 2.1 FLUID PHASE VERTICAL VELOCITY vs HEIGHT

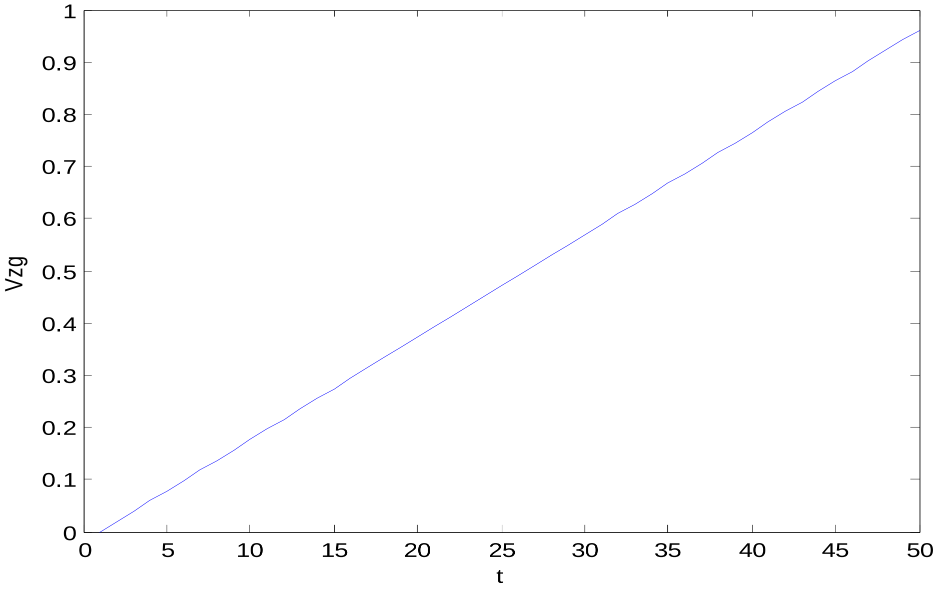


Fig 2.2 FLUID PHASE VERTICAL VELOCITY vs TIME

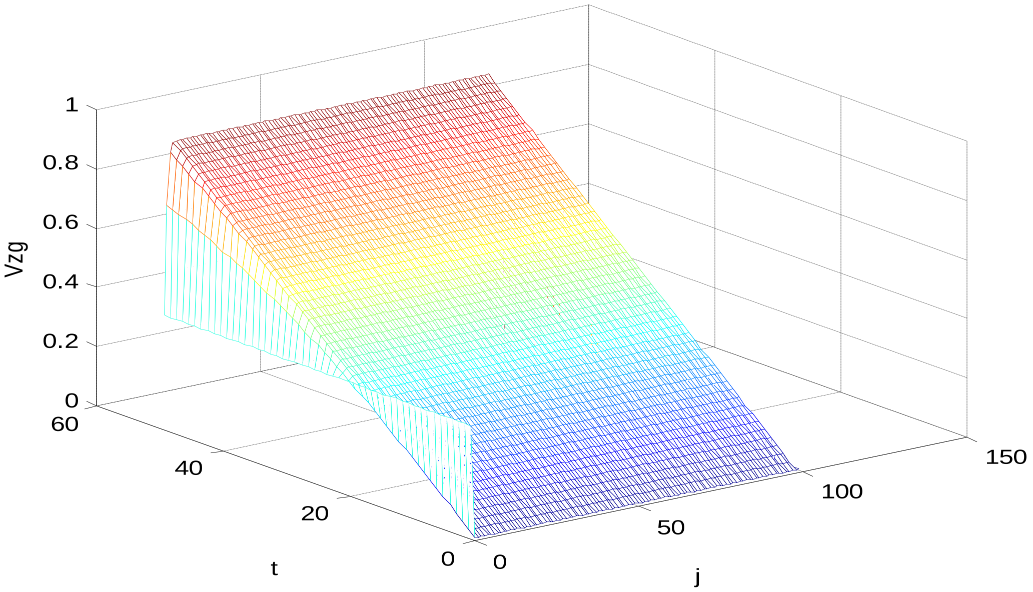


Fig 2.3 FLUID PHASE VERTICAL VELOCITY vs TIME vs HEIGHT

4.1.4 Solid phase velocity

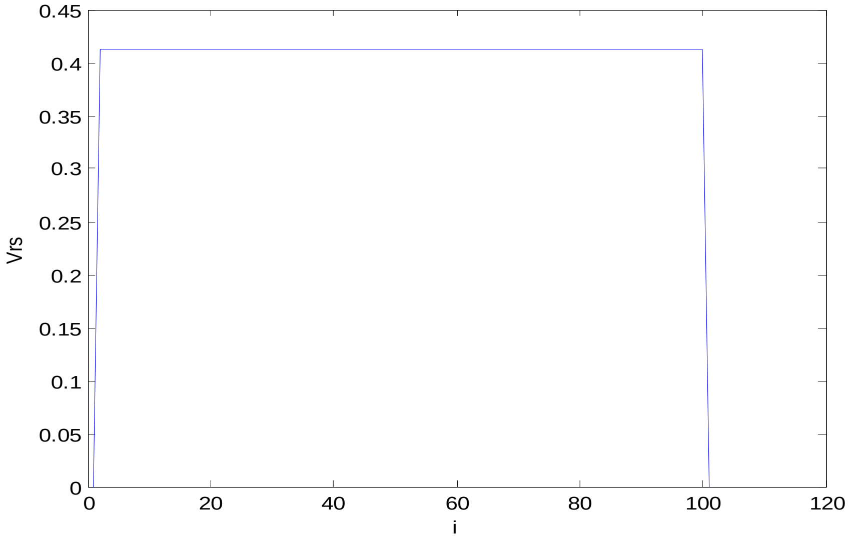


Fig 2.4 SOLID PHASE RADIAL VELOCITY vs RAIDAL DISTANCE

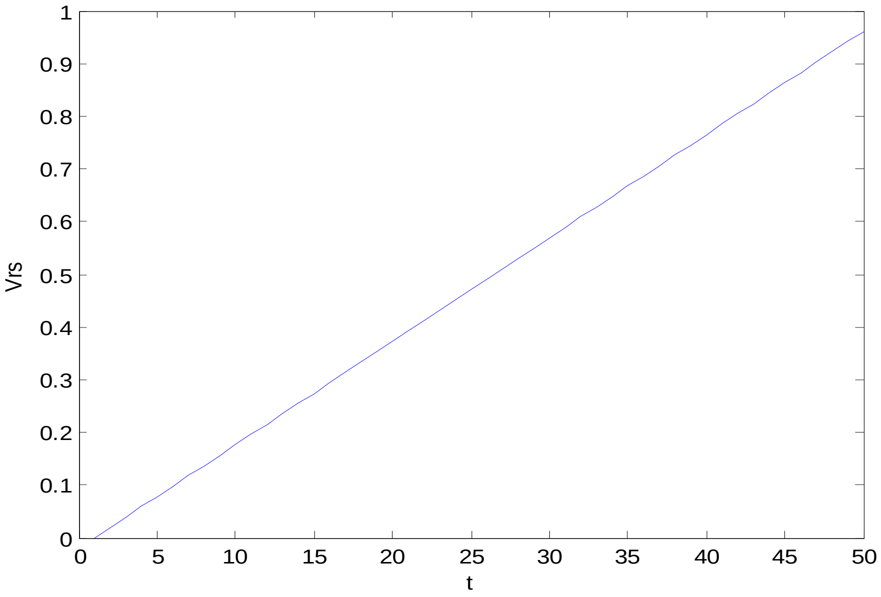


Fig 2.5 SOLID PHASE RADIAL VELOCITY vs TIME

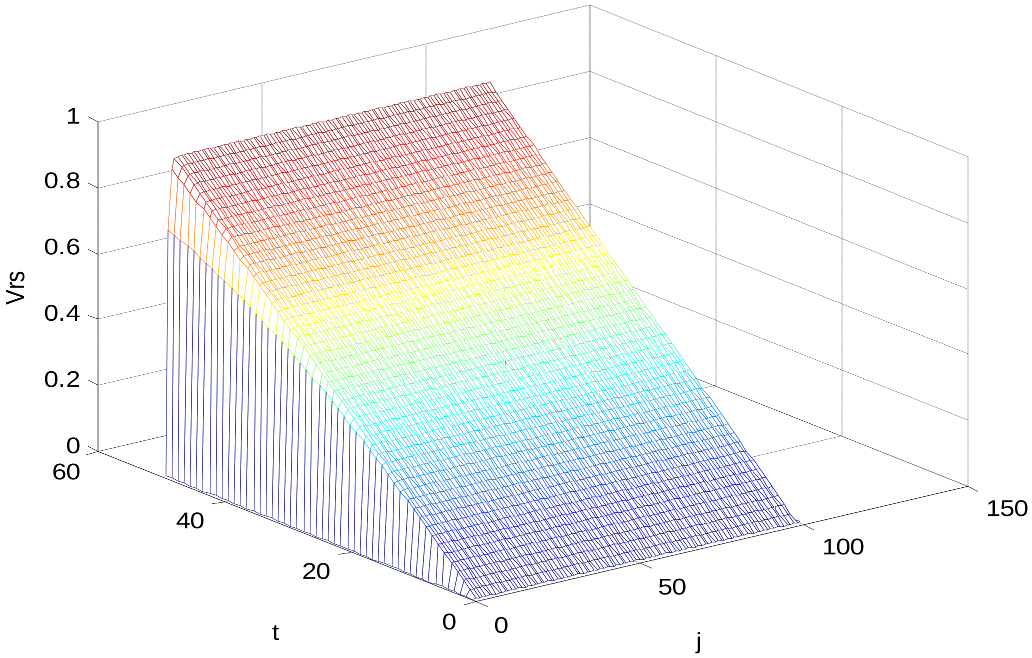


Fig 2.6 SOLID PHASE RADIAL VELOCITY vs TIME vs HEIGHT

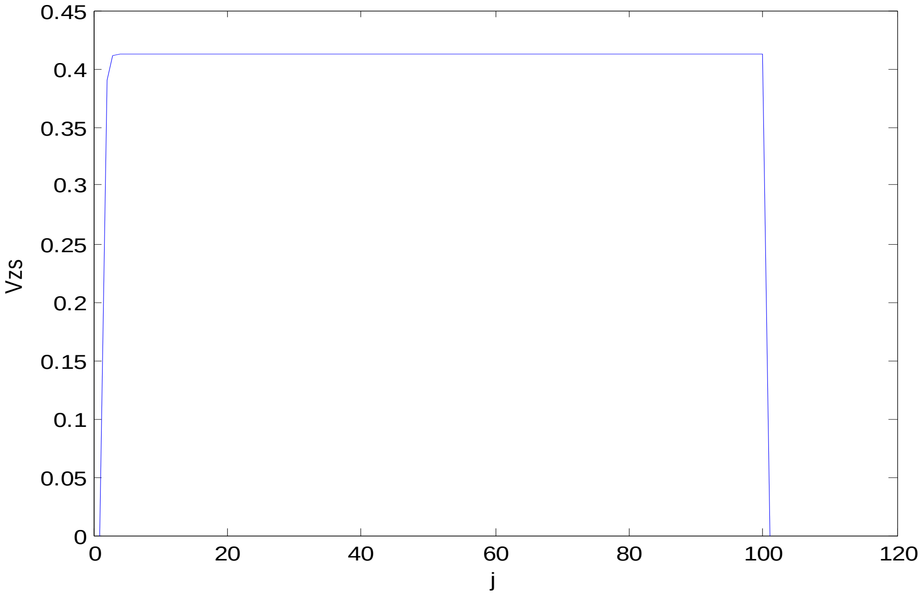


Fig 2.7 SOLID PHASE VERTICAL VELOCITY vs HEIGHT

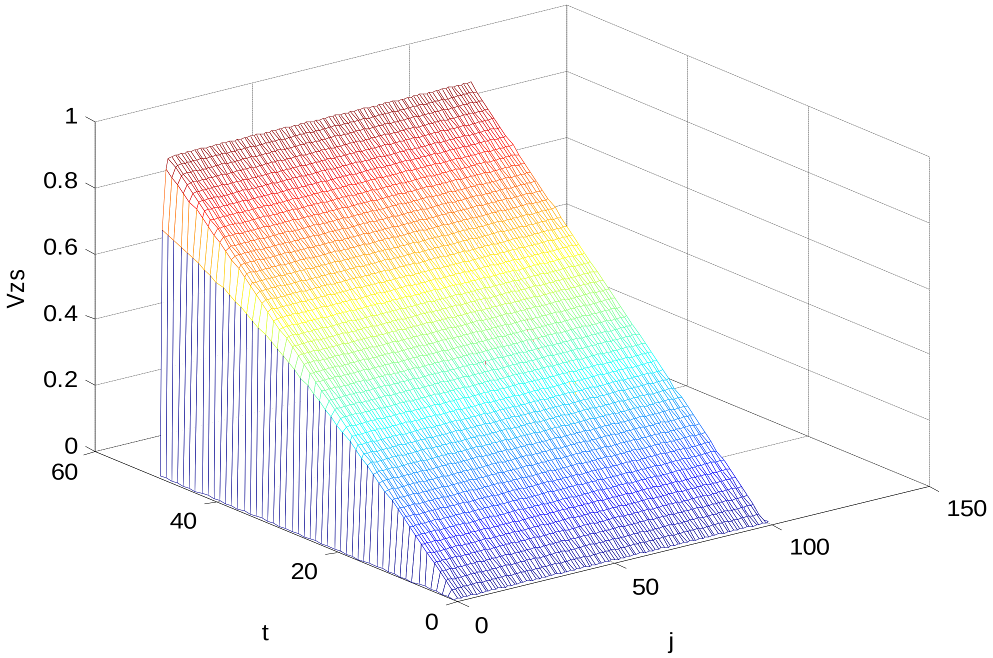


Fig 2.8 SOLID PHASE VERTICAL VELOCITY vs HEIGHT vs TIME

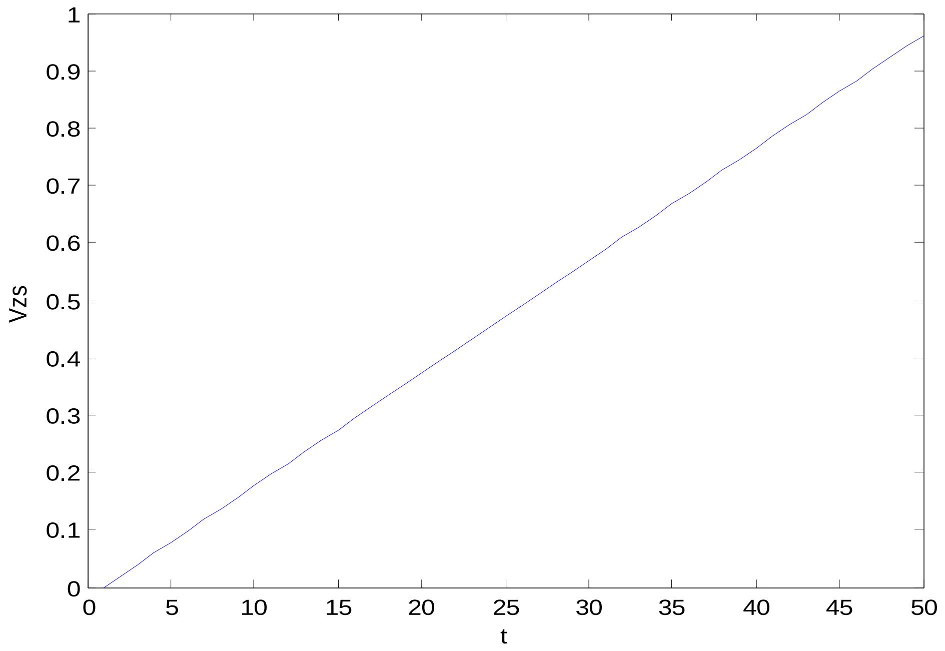


Fig 2.9 SOLID PHASE VERTICAL VELOCITY vs TIME

4.2 CONCLUSION

Hydrodynamics of the fluidized bed is studied with the help of computational fluid dynamics (CFD) techniques. Volume fraction of solid phase and liquid phase are studied and a volume fractions of solid phase and liquid phase are calculated at each point of the fluidized bed. Since in the industry we deal with the different type of particles having different properties like particle diameter , density , viscosity etc. Here in this work a code is developed which can be used for a range of particles or for a range of properties. So we can calculate the properties like volume fractions , velocity inside the fluidized bed etc for a range of particle diameter and range of density . Here in this results obtained are not satisfactory and improvement in needed in the algorithm of the code.

NOMENCLATURE :-

CD_S	Single particle drag function
d_P	Diameter of the particles
F_{gs}	Coefficient for the interphase force between the fluid phase and solid phase
I	Identity tensor
Pr	Prandtl number
f_{gs}	Momentum transfer from fluid phase to solid phase
k_S	Solid phase conductivity
P	Pressure
Re	Reynolds number
V	Velocity
r	Radius of fluidized bed
z	Height of fluidized bed

Greek symbols

v_r	Ratio of the terminal velocity of particles
Є	Volume fraction
μ	Viscosity
ρ	Density
σ	Stress tensor

Subscripts

g	Index for gas phase
s	Index for the solid phase

REFERENCES :-

[1] D. Kunii, O. Levenspiel, Fluidization Engineering, Second ed. Butterworth-Heinemann, Boston, 1991.

[2] D. Gidaspow, Multiphase Flow and Fluidization, First ed. Academic press, London, 1994.

[3] V.V. Ranade, Computational Flow Modeling for Chemical Reactor Engineering, First ed. Academic press, New York, 2002.

[4] F. Taghipour, N. Ellis, C. Wong, Experimental and computational study of gas–solid fluidized bed hydrodynamics, Chemical Engineering Science 60 (2005) 6857–6867.

[5] M.J.V. Goldschmidt, R. Beetstra, J.A.M. Kuipers, Hydrodynamic modeling of dense gas-fluidized beds: comparison and validation of 3D discrete particle and continuum models, Powder Technology 142 (1) (2004) 23–47.

[6] L. Huilin, H. Yurong, D. Gidaspow, Hydrodynamic modeling of binary mixture in a gas bubbling fluidized bed using the kinetic theory of granular flow, Chemical Engineering Science 58 (7) (2003) 1197–1205.

[7] B.G.M. Van Wachem, J.C. Schouten, C.M. Van den Bleek, R. Krishna, J.L. Sinclair, Comparative analysis of CFD models of dense gas–solid systems, AIChE Journal 47 (5) (2001) 1035–1051.

[8] F. Rong, D.L. Marchisio, R.O. Fox, CFD Simulation of Polydisperse Fluidized-Bed Polymerization Reactors, Department of Chemical Engineering, Iowa State University, 2114 Sweeney Hall, Ames, IA 50010-2230, USA, Preprint submitted to Elsevier Science, August 2003.

[9] C. Pain, S. Mansoorzadeh, C.R.E.D. Oliveira, A.J.H. Goddard, Numerical modeling of gas–solid fluidized beds using the two-fluid approach, International Journal for Numerical Methods in Fluids 36 (2001) 91–124.

[10] B.G.M. Van Wachem, J.C. Schouten, C.M. Van den Bleek, R. Krishna, J.L. Sinclair, CFD modeling of gas-fluidized beds with a bimodal particle mixture, AIChE Journal 47 (6) (2001) 1292–1302.

[11] S. Cooper, C.J. Coronella, CFD simulations of particle mixing in a binary fluidized bed, Powder Technology 151 (2005) 27–36.

[12] Y. Kaneko, T. Shiojima, M. Horio, DEM simulation of fuidized beds for gas-phase olefn polymerization, Chemical Engineering Science 54 (1999) 5809–5821.

[13] A. Gobin, H. Neau, O. Simonin, J. Llinas, V. Reiling, J.L. Selo, Fluid dynamic numerical simulation of a gas phase polymerization reactor, International Journal for Numerical Methods in Fluids 43 (2003) 1199–1220.

[14] L. Huilin, L.Wentie, L. Feng, Z. Guangbo, H.L. Huilin, L.Wentie, L. Feng, Z. Guangbo, H. Yurong, Eulerian simulations of bubble behavior in a two-dimensional gas–solid bubbling fluidized bed, International Journal of Energy Research 26 (15) (2002) 1285–1293.

[15] M. Chiesa, V. Mathiesen, J.A. Melheim, B. Halvorsen, Numerical simulation of particulate flow by the Eulerian–Lagrangian and the Eulerian– Eulerian approach with application to a fluidized bed, Computers & Chemical Engineering 29 (2) (2005) 291–304.

[16] E.J.G. Eriksson, T.F. McKenna, Heat-transfer phenomena in gas-phase olefin polymerization using computational fluid dynamics, Industrial and Engineering Chemistry Research 43 (2004) 7251–7260.

[17] J.T. Jenkins, F. Mancini, Balance laws and constitutive relations for plane flows of a dense, binary mixture of smooth, nearly elastic circular disks, Journal of Applied Mechanics 54 (1987) 27–43.

[18] D. Gidaspow, L. Huilin, E. Magner, Kinetic theory of multiphase flow and fluidization: validation and extension to binaries. 19th International Congress of Theoretical and Applied Mechanics, Kyoto, Japan, 1996.

[19] Y.L. Shi, A.S. Mujumdar, M.B. Ray, Heat transfer in multiple-turbulent-slot impinging jets of gas–particle suspensions, Numerical Heat Transfer. Part A, Applications 47 (2) (2004) 147–164s.

[20] N.J. Mariani, S.P. Bressa, G.D. Mazza, O.M. Martínez, G.F. Barreto, A BEM formulation for evaluating conductive heat transfer rates in particulate systems, Numerical Heat Transfer Part B, Fundamentals 31 (4) (1997) 459–476.

[21] S.V. Patankar, Numerical heat transfer and fluid flow, First ed. Hemisphere Publishing, Washington, DC, 1980.

[22] Bird R.B. , Stewart W.E.,Lightfoot E.N. , Transport phenomena , second edition , Wiley 2002.