Numerical Simulations of Thermo-Fluid Phenomena in Microwave Heated Packed and Fluidized Beds

by

Max Savransky, M.S.

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

James R. Thomas, Jr., Chairman
Clint L. Dancey
Brian Vick
William A. Davis
Lee Johnson

September 2003

Blacksburg, Virginia

Key Words: microwave catalysis, packed beds, fluidized beds

Numerical Simulations of Thermo-Fluid Phenomena in Microwave Heated Packed and Fluidized Beds

Max Savransky, Ph.D.
Virginia Polytechnic Institute and State University, 2003
Advisor: James R. Thomas, Jr.,

Abstract

Microwave heating is implemented in various fields such as drying, material processing, and chemical reactors. Microwaves offer several advantages over conventional heating methods: 1) microwaves deposit heat directly in the material without convection or radiation, 2) microwave heating is easy and efficient to implement, and 3) microwave processes can be controlled. In order to understand how to use microwaves more efficiently, we must understand how they affect the material with which they interact. This requires the ability to predict the temperature distribution that is achieved within the material. In recent years packed and fluidized beds have been used as chemical reactors to achieve various tasks in industry. Recent studies have shown that microwave heating offers the potential to heat the bed particles to a higher temperature than that of the fluid. This results in enhanced reaction rates and improves the overall efficiency of the reactor. The focus of this work is to determine the temperature distributions within the packed and fluidized beds, and to determine whether the catalyst particles can be heated to a higher temperature than the gas in catalytic reactions. The beds are modeled with multiphase flow equations. The gas velocity profiles along with the solid and gas temperature profiles for packed and fluidized beds are provided. For the fluidized beds, the hydrodynamics is modeled using FLUENT and the solid velocity profiles are also determined.

Acknowledgments

I would like to thank my advisor, Dr. James R. Thomas, for his patience and guidance. He helped me with problems along the way. I would also like to thank Dr. Clint L. Dancey, Dr. Brian Vick, Dr. William A. Davis, and Dr. Lee Johnson for serving on my committee. Their input was very valuable. I would like to thank my family for their support through this difficult journey. I would like to thank Dr. Peter Menegay. His knowledge of computational fluid dynamics (CFD) was very valuable. I would like to thank the Dupont company and the National Science Foundation (NSF) for partial support of this work. Finally, I would like to thank Dr. Hamid Arastoopour, Dr. Dimitri Gidaspow, and their students for their input. Their knowledge of fluidization was most helpful.

Max Savransky

Virginia Polytechnic Institute and State University

September 2003

Contents

List of Figures

NOMENCLATURE

D_p	metallic catalyst particle diameter (m)
C_{pf}	specific heat for the gas phase (J/kg $\cdot$ K)
C_{ps}	specific heat for the solid phase (J/kg $\cdot$ K)
d_p	catalyst pellet diameter (m)
E	electric field (V/m)
f	microwave frequency (Hz)
g	gravitational acceleration (m/s^2)
h_v	volumetric coefficient of heat transfer (W/m^3 $\cdot$ K)
H	the height of the bed (m)
k_f	conductivity of the gas phase (W/m $\cdot$ K)
k_s	conductivity of the solid phase (W/m $\cdot$ K)
K	permeability of a porous system
L	length of the fluidized bed (m)
Nu	Nusselt number
Pr	Prandtl number
$\dot{q}$	rate of heat generated within the solid phase (W/m^3)
R	radius of the bed (m)
Re_d	Reynolds number for porous flow
T_f	temperature of the gas phase (K)
T_s	temperature of the solid phase (K)
T_∞	surrounding temperature of the bed (K)
u_D	dynamic wave velocity (m/s)
u_{Dz}	the gas velocity in a packed bed (m/s)
u_K	kinematic wave velocity (m/s)
u_{mf}	minimum fluidization velocity (m/s)
u_f	x-component of the gas velocity (m/s)
u_s	x-component of the solid velocity (m/s)
U	the overall coefficient of heat transfer (W/m^2 $\cdot$ K)
v_f	y-component of the gas velocity (m/s)

v_s	y-component of the solid velocity (m/s)

Greek Symbols

ϵ_0	permittivity of free space (f/m)
ϵ	electric permittivity of materials (f/m) or emissivity
ϵ'	dielectric constant (f/m)
ϵ''	loss factor (f/m)
ϵ''_r	effective relative loss factor
μ_0	permeability of free space (h/m)
μ	dynamic gas viscosity ($\mathrm{N \cdot s/m^2}$)
ρ_f	density of the gas phase ($\mathrm{kg/m^3}$)
ρ_s	density of the solid phase ($\mathrm{kg/m^3}$)
σ	electrical conductivity (S/m)
ϕ_s	sphericity coefficient
ϕ_0	porosity at the center of the bed
ϕ	porosity in a packed bed
ϕ_f	volume fraction of the gas in a fluidized bed
ϕ_s	volume fraction of the solid in a fluidized bed
Φ	volume fraction of catalyst within the pellet

Chapter 1

Introduction

1.1 Prelude

The utilization of microwave energy for heating processes may offer some advantages over conventional heating methods. For dielectric and semi-conducting materials, microwave heating results in better material properties, and time and energy savings. This is because microwaves deposit energy directly in the medium as opposed to conventional heating where heat is delivered by a convective or conductive process. Besides material processing, microwave heating is also utilized in the enhancement of chemical reactions. In some instances, a material that can absorb microwave energy is used to transfer the energy to the reactant because many chemicals cannot absorb the microwave energy directly.

An area where microwave heating offers the advantage of direct heat deposit is chemical catalysis. Several researchers have found that microwave heating brought about an increase in reaction rates and yield over conventional heating. Bond et al. [1] reported that microwave heating reduced catalyst drying time by at least a factor of two while improving their crushing strength. Chemat et al. [2] applied microwave heating to esterification and found increases in reaction rates compared to conventional heating. Kim et al. [3] used a microwave heated fluidized bed reactor for fast pyrolysis of chlorodifluoromethane into tetrafluoroethylene and found that they could achieve yields of more than 80 percent over a temperature range of 700-840 C.

Berlan et al. [4] found that reaction rates in cycloaddition reactions were faster when heated by microwaves than with conventional heating.

Two theories have been suggested as explanations of these phenomena. The first suggests that an increase in the pressure or temperature in the reactor brings about the influence of thermal effects. This is a result of the heat generation that is caused by the interaction between the electromagnetic field and the medium. This interaction may lead to "hot spots" because the electromagnetic field may be locally concentrated in and around the metallic catalyst particles without any noticeable change in the bulk temperature of the system. The second theory suggests the presence of "non-thermal" effects. The idea behind this theory was that microwaves would couple with the desired molecules and alter their rotations, thus increasing yields of the reactions. Stuerga and Gaillard [5] opposed this explanation by showing that such effects are improbable. Despite this fact, many papers have been published successfully demonstrating such effects. In reality, it may be a combination of both that is contributing to the enhancement of the reaction rates, but very little is understood about the microscopic nature of "non-thermal" effects because measurements are very difficult to perform at such a level. This inability in making the appropriate measurements is the reason why the parameters which characterize the "non-thermal" effects have not been quantified. Thus, only the thermal effects will be considered in this work.

1.2 Objectives

A numerical investigation of gaseous flow through packed and fluidized beds in a microwave applicator will be conducted. These can be seen in Fig. 1.1. As the gas is passed through the beds, a fraction of the energy generated by the microwaves is converted into thermal energy by the pellets and transferred to the gas. The desired chemical reaction is then initiated between the gaseous molecules at some optimal temperature, T_{opt}. Side reactions may occur if the temperature is sufficiently different from T_{opt}. In order to avoid these side reactions, it would be ideal to heat the solids

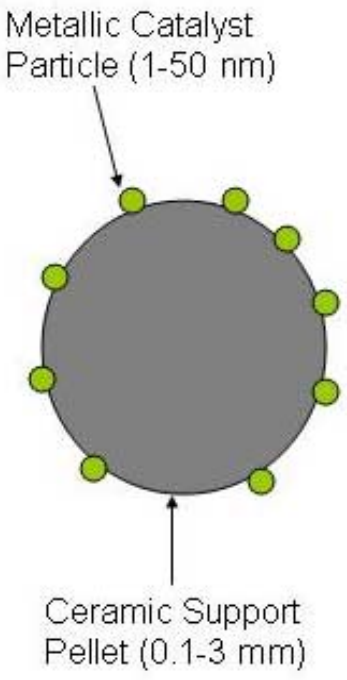


Figure 1.1: A metallic catalyst particle on a ceramic support.

to a higher temperature than the gas so that the heat supplied by the solids would maintain the gas around T_{opt}. A typical experimental arrangement for a packed bed is shown in Fig. 1.2. The fluidized bed is shown in Fig. 1.3. In this investigation, the packed and fluidized bed was composed of alumina pellets containing nm-sized platinum particles. The packed bed is approximately 1 inch in height and diameter and placed in a glass tube [6]. The fluidized bed container is about 40 cm in length and 60 cm in height and is rectangular [7].

In the packed and fluidized bed setups, we consider the flow of air over alumina/platinum catalysts. Air is a chemically inert gas. The effect of chemical reactions will not be considered because the goal of this work is to determine whether the solids can be heated to a higher temperature than the gas by microwaves. One hypothesis is that the platinum catalyst can achieve significantly higher temperatures than the ceramic support. This phenomenon is known as selective heating and is described in more detail in the section below. If this is true, then the chemical reaction

will be greatly enhanced in the vicinity of the catalyst particles. It is desired to be able to heat the particles without heating the gas so that the desired reaction will take place without any side reactions.

1.3 Selective Heating

In the application of interest in this work, metallic particles are nanometer-sized particles attached to a ceramic support. These particles are believed to couple better with the microwave energy than the support and hence achieve significantly higher temperatures, approximately $200-300°C$. The electrical conductivity becomes significantly smaller than the bulk conductivity for nanometer-sized particles, as reported by Nimtz et al. [8], allowing the microwaves to couple with the metallic catalysts. The bulk conductivity of the catalyst particles may still be higher than that of their ceramic support. For particles whose diameter is less than 500 nm, the conductivity may be represented approximately by [8]

$$\sigma_{eff} = \left(\frac{D_p}{5 \cdot 10^{-6}}\right)^3 \sigma_b, \tag{1.1}$$

where D_p is the particle diameter (m) and σ_b is the bulk conductivity of the metal. The catalyst particles thus behave as dielectrics.

1.4 Numerical Model

A thermal model assuming a constant and attenuated electric field in the packed and fluidized beds is developed. The thermal model for the packed bed and fluidized beds provide temperature distributions for the bed particles. The packed bed model also provides the velocity profile of the gas, but the velocity profiles are obtained with the aid of FLUENT [9], a commercial computational fluid dynamics (CFD) software package, for the fluidized bed. The velocity profiles determine the rates at which the gas removes heat from the particles and the electric field determines the rate of heat generation within the particles. The temperature disribution of the gas will

determine the extent to which the reaction will be enhanced. Also, the temperature difference between the gas and solid phases will serve as an indicator of whether selective heating is likely to occur. A moderate temperature difference between the gas and solid indicates that selective heating is possible while a very small temperature difference does not reveal anything about selective heating.

Thus far, microwave effects on packed and fluidized beds have been studied by Lanz [10] and Faucher [11] respectively. Both have developed one-dimensional heat transfer models to analyze the temperature distributions and both have assumed constant gas velocity and zero solid velocity. Hence, Faucher's model does not account for the motion of the solids and the expansion of the bed while Lanz's model does not account for the radial variation in gas velocity which can produce temperature gradients in that direction. These limitations are removed in the current model.

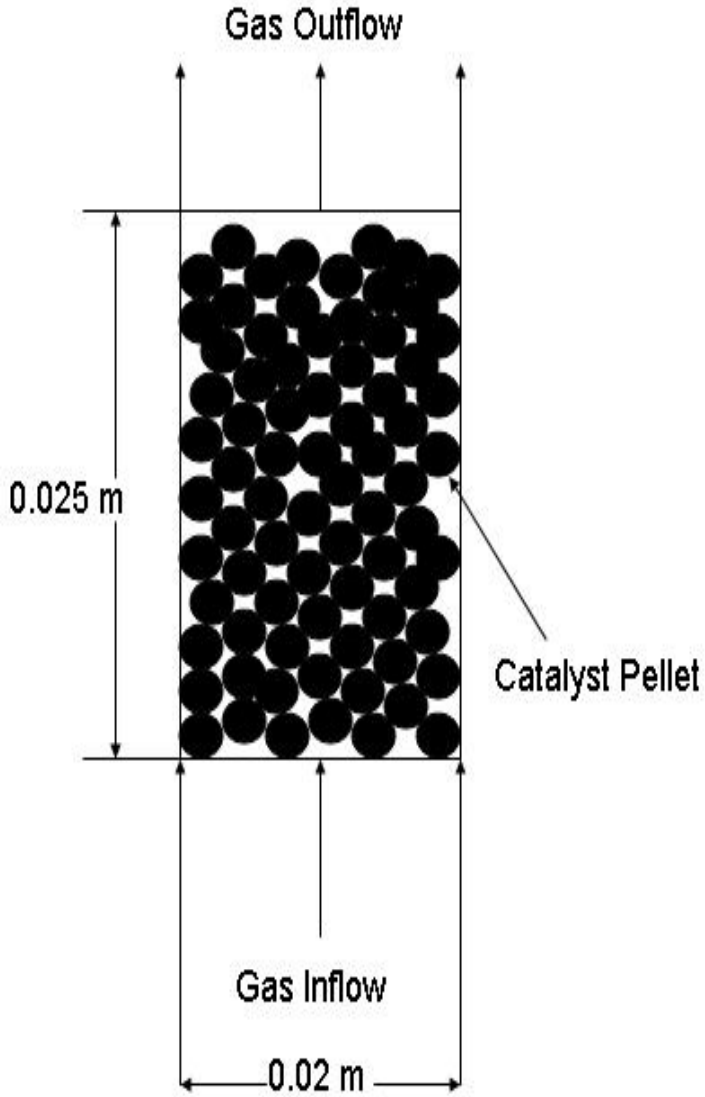


Figure 1.2: Packed bed setup.

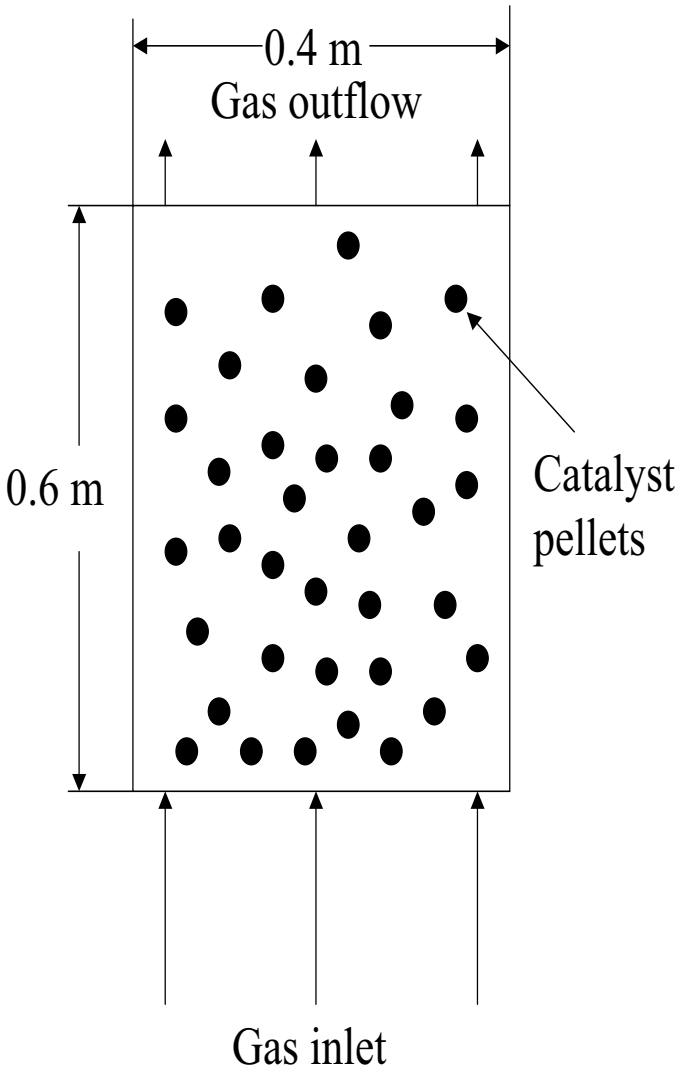


Figure 1.3: Fluidized bed setup.

Chapter 2

Literature Review

2.1 Increased Reaction Rates Via Hot Spots

Several researchers have concluded that the enhancement of reaction rates under microwave irradiation is due to the presence of "hot spots". Chemat et al. [2] deduced the existence of a temperature difference in liquid phase chemical reactions under the influence of microwaves through thermodynamic calculations. They used 3 to 5 mm montmorillonite clay pellets with $Fe_2(SO_4)_3$ catalyst particles. The temperature differences between the reactant and the catalyst pellets were estimated to range from 9-18 C. It was concluded that the temperature difference and the increase in yields were due to microwave heating.

Using a lumped-capacitance model and a kinetic theory description of the heat transfer coefficient, Thomas [12] showed that the catalyst particles can be heated to higher temperature than the support into which they are imbedded when the gas pressure is reduced to 0.1 atm.

2.1.1 Increased Selectivity

The investigation of chemical reactions under the influence of microwaves has been studied since the 1970s. As mentioned earlier, microwaves have the advantage of delivering heat directly within the catalyst as opposed to conventional heating where

the heat is delivered to the catalyst by a heat exchanger by convection and possibly radiation. Several authors reported increased selectivity when the reactor was heated by microwaves.

Ioffe et al. [13] emphasized that irradiation time is crucial to the observed increases in selectivity. He used radio frequency and microwave heating to produce acetylene from methane and hydrogen.

Roussy et al [14] reported much higher selectivities in isomerization of 2-methylpentane. However, the higher yields were due to subjecting the catalyst to an electromagnetic field prior to utilization in a chemical reactor. Also, the nature of the catalyst particle added to the γ - alumina played an important role in the increased selectivity. Wan [15] used microwave heating to produce hydrocarbons from water and carbon. He emphasized the importance of the catalysts used in the reaction and observed that catalysts serve a dual purpose; 1) increase the chemical reactivity, and 2) convert microwave energy into chemical energy.

2.1.2 Modeling Efforts

Multiphase flow problems arise in many areas in engineering such as chemical reactors, conveyers of solid materials, and heat exchangers. Therefore, correct prediction of the flow and thermal behavior in these various devices is important in their design. Several approaches have been used to solve the fluid mechanics portion of the multiphase flow problem.

The first approach is based on empirical methods [16,17]. The resulting equations derived based on this approach are restricted in their application by the parameters of the particular system to which they are applied.

A second approach is based on the continuum mixture theory [18,19] in which the phases are assumed to exist as overlapping continua over the entire region of interest. Each variable is expressed as the sum of the volume fractions of the variable for each phase. Only a single conservation of mass, momentum, and energy equation is written to describe the state of the system.

A third approach which has been utilized with much success by Kawaguchi

et. al [20] is the discrete element method (DEM). It is assumed that the fluid phase is viewed as a continuum while the particles are treated discretely. The fluid phase variables are described by volume-averaged continuity and Navier-Stokes equations while the state of the solid phase is governed by Newton's laws. The relationship for the drag force between the solid and fluid states is determined empirically.

Kawaguchi used the DEM to simulate the behavior in a two-dimensional fluidized bed in which particle motion was restricted by parallel plates inserted inside the bed. The disadvantage of this approach is that it is difficult to implement it to fine-particle systems because the computational capabilities are insufficient for this method to be applied to systems with a large number of particles.

A fourth approach is based on the method of local volume averaging [21]. Each phase is treated as an interpenetrating continuum and occupies a different portion of the region of interest. Each phase is also assumed to be separated by irregular surfaces. Conservation laws are written for each phase with appropriate boundary conditions at the interphases. However, since the boundaries undergo changes in their size and shape with space and time, solving these equations is an overwhelming task. Hence, the field variables are averaged over a representative control volume which is large in comparison to the control volume for a single phase and yet small in comparison to overall dimensions of the region of interest. A resulting set of differential equations is developed for each phase.

Local volume averaging was first employed by Anderson and Jackson [22] to describe the fluid mechanics in a fluidized bed. Drew and Segel [23] applied the same technique to study fluidized beds and foams. Hassanizadeh and Gray [21] developed a generalized approach to multiphase flow systems without considering any particular applications. Bogere [24] later applied a more rigorous form of the generalized approach to derive the governing equations in gas-solid fluidized beds. Gidaspow et. al [25,26] used volume-averaged Navier-Stokes equations to analyze bubbling flow in fluidized beds. They also validated the kinetic theory approach which is used to determine the solids particle viscosity. Arastoopour et. al [27,28] used the volume-averaging capabilities of FLUENT and CFX, another CFD software package, to study

the solids motion in circuilating fluidized beds (CFB).

Another possible approach is to compute the desired quantities directly without approximation at each location and at each time step in the system. One possible way to accomplish this would be to combine the portion of the DEM method for the solid phase used in [20] with the direct solution of the Navier-Stokes equations for the fluid on a domain which consists of the interior of the container with volumetric voids induced by the occupation of the solid particles. The DEM method would then provide the location and the velocity of the solid particles which could then be used to identify the geometry of the fluid and the boundary condition at the interface between the solid particle and the gas for the solution of the Navier-Stokes equations. The magnitude of the x, y, and z-component of velocity of the solid particle is the boundary condition which is implemented in the x, y, and z-component of the Navier-Stokes equations respectively. The problem with this method is that when the particles change location, a new grid is required for the solution of the Navier-Stokes equations which means that a new grid is required at every time step of the solution. Such a method is impractical with current computational capabilities.

Several researchers have also used the first four of the above approaches to study the thermal problem in packed and fluidized beds. Thomas and Faucher [29], used the volume-averaged energy equations to study packed and fluidized beds. The packed bed was modeled by an approach which was based on the work of Choudhury [30] while the fluidized bed was modeled in a similar manner except with a different correlation for the heat transfer coefficient between the solid and gaseous phases. Zahavi [31] used the energy equations to study liquid-solid fluidized beds. He considered the motion of the solid particles but used a mixture approach instead of the two-phase approach used by Thomas and Faucher. Chen and Wang [32] derived a simplified one-dimensional steady-state model for a gas-solid fluidized bed. They wanted to study the effects of the fluidization on the temperature and saturation within a single particle. The particle was assumed to be represenstative within the fluidized bed. The governing differential equations with the appropriate boundary conditions for the temperature and saturation were provided. The same authors [33] used a

similar approach to study fluidized bed drying with microwave heating. The heat generation within a particle was modeled as a constant times the square of the electric field. The electric field was assumed to be constant. Minoura et al. [34] used the volume-averaged energy and mass transfer equations to develop a three-dimensional multiphase model which describes the temperature and oxygen concentration distributions in a fluidized bed refuse incinerator. However, they assumed constant velocities for all phases. Roussy et. al [35] developed a two-phase two-dimensional model to predict the heat transfer in a fluidized bed which was placed in a multimode rectangular cavity. They assumed the particles were stationary and that the gas velocity was constant. Also, the model is one-dimensional in each phase. They found that the electric field is strongly dependent on the motion of the solid in a multimode cavity.

Several organizations have conducted experiments on microwave-heated fluidized beds [36,37]. In their setups, the solids are initially fed into the bed until the desired solid capacity is attained. The solids entrance valves are then closed and the measurements are taken. Unfortunately, the results presented in these papers are very difficult to compare with the current model since the geometry of the system is different and not all of the necessary input data is provided.

Chapter 3

Interactions Between Materials and Electromagnetic Waves

3.1 Introduction

Microwave processes involve the generation and transmission of electromagnetic energy to some particular system. The microwave system consists of three main components; the generator, the transmission line, and the applicator [6]. In low power applications, transmission lines are employed instead of waveguides. In high power applications, waveguides are preferred because transmission lines yield large conductor losses.

Several devices are employed as microwave generators. Some of the common sources include traveling wave tubes (TWT), Gunn diodes, klystrons, and magnetrons. In most microwave applications, the generator is a magnetron tube which consists of a cathode/anode system. The cathode is heated to a sufficiently high temperature and then set to a higher negative potential than the anode. The electrons are boiled off from the cathode and propagate towards the anode in a spiral, thus generating an electric field. The magnetron is simple, reliable, efficient, and cost-effective as opposed to the other devices. Typical commercial magnetrons are designed to be operated in a TE_{10} launcher [1].

[1]This notation describes the specific electromagnetic field mode in the waveguide. Additional

A waveguide provides a coupling between the generator and the applicator. It is a long hollow conductor of constant cross-sectional dimension such that its dimensions are at least a quarter of the operational wavelength. This size is substantially larger than that of a comparable transmission line, and hence the losses will be drastically reduced. The appropriate electric field determines the dimensions of the waveguide. Rectangular waveguides are most commonly used and are usually operated in the TE_{10} mode.

An applicator is needed to produce desired electromagnetic field distributions within the object to be heated. The design of the applicator is highly influenced by the dimensions of the object to be heated by the electromagnetic field. This implies that microwaves are appropriate for small objects and RF is appropriate for large ones. The frequency range is also affected by the total power to be applied. For applications in which the power is greater than 200 kW, RF systems are usually employed. For applications in which the power is between 1 and 20 kW, microwaves are preffered. For intermediate power applications, (1-100 kW), the frequency is determined by thermal and electrical properties and the geometry of the object. Hence, the applicator design is a compromise between the way in which the electromagnetic field interacts with the object to be heated and the advantages and disadvantages of different applicator-workpiece combinations.

3.2 Electric Field

The electromagnetic energy is carried by a traveling electromagnetic wave and interacts with the system in a way which is characterized by the electric conductivity, σ, the electric permittivity, ϵ, and the magnetic permeability, μ of the system. The energy can be dissipated or absorbed by the system. The system properties relate the electric field strength, through constitutive relations, to quantities which appropriately describe the relationship between the system and the electromagnetic energy.

The force exerted on a test charge Q by another point charge at rest is expressed

explanation is given in [38].

by Coulomb's Law [39],

$$\vec{\mathbf{F}} = \frac{1}{4\pi\epsilon_0}\frac{qQ}{r^2}\vec{\mathbf{r}}, \tag{3.1}$$

where ϵ_0 is the permittivity of free space, r is the distance between the two charges, and q is the magnitude of an arbitrary point charge.

Now, suppose there are n point charges $q_1, ..., q_n$, at distances $r_1, ..., r_n$ from the test charge Q. The total force on Q is the sum of the forces of the individual n point charges given by,

$$\vec{\mathbf{F}} = Q\vec{\mathbf{E}}, \tag{3.2}$$

where

$$\vec{\mathbf{E}}(\vec{\mathbf{r}}) \equiv \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{n}\frac{q_i}{r_i^2}\vec{\mathbf{r}_\mathbf{i}}. \tag{3.3}$$

Here, $\vec{E}$ is called the electric field, and is a measure of the amount of force exerted by the discrete point charges on a test charge as indicated by equation (3.2). The electric field strength is the magnitude of the electric field.

3.3 Constitutive Relations

The constitutive relations indicate that the current density, the electric flux density, and the magnetic flux density are proportional to the electric field [39].

The electric field, $\vec{E}$, is related to the electric conductivity and to the current density, $\vec{J}$, by:

$$\vec{\mathbf{J}} = \sigma\vec{\mathbf{E}}, \tag{3.4}$$

where σ and $\vec{E}$ are defined as above. The current density is measured in amperes per square meter, (A/m^2) and is a representation of the amount of current passing through a surface.

The electric field, $\vec{E}$, is related to the electric permittivity and to the electric flux density, $\vec{D}$, by:

$$\vec{\mathbf{D}} = \epsilon\vec{\mathbf{E}}, \tag{3.5}$$

where ϵ is a complex quantity whose real part represents the amount of electromagnetic energy stored within the material and whose imaginary part represents the amount of electromagnetic energy which can be converted into thermal energy and is written as:

$$\epsilon = \epsilon' - j\epsilon''. \tag{3.6}$$

The permittivity of free space is denoted ϵ_0, and its value is $8.85 \times 10^{-12} C^2/(N \cdot m^2)$. It is often convenient to represent the real and imaginary parts of the permittivity in nondimensional form relative to ϵ_0 as,

$$\epsilon'_r = \frac{\epsilon'}{\epsilon_0}, \tag{3.7}$$

and

$$\epsilon''_r = \frac{\epsilon''}{\epsilon_0}. \tag{3.8}$$

The magnetic field, $\vec{H}$, is related to the electric permeability and to the magnetic flux density, $\vec{B}$, by,

$$\vec{\mathbf{B}} = \mu\vec{\mathbf{H}}, \tag{3.9}$$

where μ is a complex quantity whose real part represents the amount of magnetic energy stored within the material and whose imaginary part represents the amount of magnetic energy which can be converted into thermal energy and is written as,

$$\mu = \mu' - j\mu''. \tag{3.10}$$

3.4 Dielectric Losses

Dielectric materials can be classified as semiconductors. The atomic and molecular forces which bound the electrons in these materials are sufficient to conduct some electricity but not to the same extent as in metals. When these materials are irradiated by an electric field, their dominant electric charges tend to align with the field.

The direction of the field turns out to be the vector from the negative to the corresponding positive charges. There is an energy loss due to interatomic friction forces which results during this alignment. If the field is alternating, the electron alignment is also alternating, thus causing electric energy to be converted into molecular kinetic energy.

In general, it is the valence electrons of the material that are most affected by the electric field. These electrons can exist only in discrete energy levels. In order for them to pass from one energy level to another, they must emit or absorb an exact amount of energy. The applied electric field supplies the energy which is required to alternate the valence elctrons from one state to another. Materials which couple strongly with the electric field are called lossy materials. The electric energy within the material is dissipated as thermal energy and the dissipation is described by the dielectric loss of the material.

The power absorbed by an object is the difference between the entering and exiting electromagnetic fields and is given by [6]

$$-\int_{\mathbf{S}} \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{E}\times\mathbf{H}^*)\mathbf{ds} = \int_{\mathbf{V}} \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{E}\cdot\mathbf{J})\mathbf{dV} + j\omega\int_{\mathbf{V}} \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{B}\cdot\mathbf{H}^* - \mathbf{E}\cdot\mathbf{D}^*)\mathbf{dV}. \tag{3.11}$$

where S is the surface area of the object, V is the volume of the object, and * indicates complex conjugation. Applying the constitutive relations to equation (3.11) yields

$$\begin{aligned} -\int_{\mathbf{S}} (\mathbf{E}\times\mathbf{H}^*)\mathbf{dS} \;&=\; \int_{\mathbf{V}} \sigma|\mathbf{E}|^{\mathbf{2}}\mathbf{dV} + \frac{\mathbf{1}}{\mathbf{2}}\omega\int_{\mathbf{V}} (\epsilon''|\mathbf{E}|^{\mathbf{2}} + \mu''|\mathbf{H}|^{\mathbf{2}})\mathbf{dV} \\ &\quad +\mathbf{j}\frac{\mathbf{1}}{\mathbf{2}}\omega\int_{\mathbf{V}} (\mu'|\mathbf{H}|^{\mathbf{2}} - \epsilon'|\mathbf{E}|^{\mathbf{2}})\mathbf{dV}. \end{aligned} \tag{3.12}$$

The first integral on the righthand side represents the amount of energy dissipated due to conduction losses while the second integral on the right represents the amount of energy dissipated due to dielectric and magnetic losses. The final integral represents the net amount of electric and magnetic energy stored, but does not cause any heating of the object. Equation (3.12) is a generalization of the power loss that occurs within the material. Some simplifications will be made to describe the heat generation accurately without resorting to the complexities of implementation of equation (3.12).

In the development of a model for the heat generation, it is assumed that the magnetic permeability contributes very little in comparison with the electric permittivity to the amount of heat dissipated inside the material. For our purposes, the losses due to conduction are also negligible and only the second term on the right side of equation (3.12) is important. Hence, equation (3.12) reduces to

$$P_{ave} = \omega \epsilon_0 \epsilon_r'' E_{rms}^2 V, \tag{3.13}$$

where ω is the frequency, V is the volume of the material, and E_{rms}^2 is the root mean square of the electric field, or $1/\sqrt{2}$ times the peak value. The heat generated per unit volume within the material can now be determined by substituting $\omega = 2\pi f$, and dividing both sides of equation (3.13) by V giving:

$$\dot{q} = 2\pi f \epsilon_0 \epsilon_r'' E_{rms}^2. \tag{3.14}$$

This is used to determine the heat generation within the beds.

3.5 Effective Permittivity

When metallic catalyst particles are attached to a ceramic support, the overall electric properties are altered. Several mixture formulae are available to account for the addition of the catalyst. We would like to be able to determine the permittivity of the mixture so that equation (3.14) can be used to compute the heat generation per unit volume. In particular, we use the Rayleigh mixture formula [6] to compute the *effective* permittivity.

For small spherical inclusions within a ceramic support, the permittivity is estimated by [6]

$$\epsilon_1 = \epsilon' - j\frac{\sigma_2}{\omega}, \tag{3.15}$$

and for small conducting spheres, the permittivity is estimated with

$$\epsilon_2 = \epsilon_0 - j\frac{\sigma_2}{\omega}, \tag{3.16}$$

where σ_2 is the size-dependent conductivity discussed in the first chapter. The effective permitivity of the support/catalyst mixture, where the support is a continuous medium with dispersed spherical catalysts is given by

$$\epsilon_m = \frac{\epsilon_1(2\epsilon_1 + \epsilon_2) + 2\Phi\epsilon_1(\epsilon_2 - \epsilon_1)}{2\epsilon_1 + \epsilon_2 - \Phi(\epsilon_2 - \epsilon_1)}, \tag{3.17}$$

where Φ is the volume fraction of catalyst within the support/catalyst system.

3.6 Maxwell-Wagner Effect

Real materials contain impurities which need to be accounted for when calculating the electric field within the materials. The presence of conducting impurities within a dielectric material is explained by the Maxwell-Wagner effect. When an external electric field is applied to the material, the charge distribution around the material is non-uniform. The time in which the charge density decays to 1/e of its original value is called the time constant and is given by:

$$\tau = \frac{\epsilon'}{\sigma}. \tag{3.18}$$

For a perfectly conducting sphere, the charge time constant approaches 0 and so a surface charge is produced almost immediately. In a semiconducting sphere immersed within a dielectric and irradiated by an external step electric field, an initial electric field exists inside the sphere and is given by:

$$E_2 = \frac{3\epsilon_1}{2\epsilon_1 + \epsilon_2} E_0, \tag{3.19}$$

where E_0 is the applied electric field. The surface charge around the sphere is initially zero because the charge has not had sufficient time to respond to the external field. As time progresses, a surface charge arises due to free charges within the semiconductor. The charge keeps moving until the internal electric field is zero. Thus steady-state is reached. In general, the internal field is a function of the electric properties of the semiconductor, as well as the spherical dimensions and the frequency of the applied electric field.

Chapter 4

Packed Beds

4.1 Porous Media

A permeable solid, or porous medium, is a material which consists of a solid matrix with an interconnected void [1]. Many important industrial applications such as chemical reactors, groundwater flow, and gas flow in reservoirs require an understanding of heat and fluid flow within a permeable solid. The permeability is a measure of the flow conductance of a solid matrix which accounts for hydrodynamic characteristics such as the interstitial surface area and fluid particle path, and has the units m^2. A more precise definition is given in the section below.

Porous media can be natural or artificial. The flow through a porous medium can be viewed as flow through conduits, or *pores*. The pore distribution within a natural porous solid is highly irregular. An analysis of flow phenomena is complicated due to geometrical considerations. Thermo-fluid quantities are highly irregular near the pore region. However, experimental quantities are measured over surfaces which coincide with a sufficiently large number of pores. Hence, more regular changes with respect to space and time are observed in the thermo-fluid quantities and a theoretical analysis based on plain media becomes possible.

In a porous medium, the pore sizes may vary from molecular size to the order of centimeters and the system dimensions are usually much larger than that of the

[1] Most of the discussion in this chapter is based on Ref. [40].

pores. In the case where the system dimensions are comparable to those of the pores, it is common to perform the analysis for a small number of pores.

The voids in a porous medium can be interconnected, dead end, or isolated. Dead end voids are those in which the void is connected to only one other pore. The porosity, ϕ, is defined as the fraction of the total volume of the porous medium which is occupied by the voids. Flow phenomena may only occur within an interconnected void system. The volume fraction of the interconnected pores is called the *effective porosity*. In a porous solid, the difference between the effective porosity and the porosity can be significant. In a loose medium such as packed beds, the porosity and the effective porosity are the same. The pore length scale is an important parameter in characterizing a porous matrix. Since pore geometries are highly irregular, defining the pore length is a complicated task. The *distributed linear scale* for each void is typically used in defining the pore length. Here, the linear scale is the diameter of a sphere that occupies the cross section at an arbitrary location along the pore. The average pore diameter is then determined from a statistical average of the distribution of length scales. The nonuniformities around the pore boundaries have a significant influence on the transfer rates at the boundaries and require local porosity distributions. The pore velocity, u_p, is the average velocity in a conduit.

Dybbs and Edwards[41] define a Reynolds number which is based on the average pore velocity, $\overline{u_p}$ and an average pore length scale, $\overline{d}$,

$$Re_d = \frac{\rho \overline{u_p} \overline{d}}{\mu}, \tag{4.1}$$

where ρ is the density of the fluid and μ is the dynamic fluid viscosity. They defined four flow regimes based on this Reynolds number:

- $Re_d \leq 1$, creep flow regime:

  In this regime, the viscous effects are dominant, and the flow is influenced by the pore geometry. Channeling is found to occur at the wall because the particle packing is much more loose in that region of the bed. The velocities adjacent to the walls may be nearly twice that found in the rest of the bed. The effects

due channeling propagate up to two particle diameters from the wall. The flow is fully developed at about three particle diameters from the entrance region.

- $1 \leq Re_d \leq 150$, inertial flow regime:

  When $1 \leq Re_d \leq 10$, both viscous and inertial forces are important. This is the most difficult problem to solve. For flows where $10 \leq Re_d$, the boundary layer thickness decreases compared to flows with $Re_d < 10$, due to increasing dominance of the inertial effects. The pressure drop becomes dependent on the lateral and longitudinal pore dimensions as the flow develops in each pore.

- $150 \leq Re_d \leq 300$, unsteady laminar flow regime:

  In this regime, oscillations become apparent. The observed frequencies are on the order of 1 Hz while the amplitudes are approximately one-tenth of a particle diameter. The flow remains laminar with a wake instability causing the flow to become unsteady.

- $300 < Re_d$, unsteady turbulent flow regime:

  A turbulent dispersion of injected dye occurs in this regime. The normalized pressure drop becomes constant with increasing Re_d.

The flow development through pores in the inertial flow regime is significant. As the diameter-to-length ratio approaches unity, the average pressure drop becomes much larger than that in fully developed flows. In the following sections a development of the drag force exerted by the gas on the porous system is presented. The expression for the drag force is used in the Navier-Stokes equations to describe the velocity profile of the gas.

4.2 Darcy's Law

In 1856, Henri Darcy devised an experiment to relate the pressure drop in a pipe to the *filter* velocity, u_D. In his experiment, Darcy partially filled a pipe with randomly and loosely packed uniform-size particles. The flow within the pipe was steady and

one-dimensional and the filter velocity was determined by dividing the mass flow rate by the product of the fluid density and the cross-sectional area of the voids of the porous particle system. The filter velocity and the pore velocity are related by

$$u_D = \phi u_p. \tag{4.2}$$

Darcy found the relation

$$\nabla P = -\frac{\mu}{K} u_D, \tag{4.3}$$

where K is the permeability matrix, ∇P is the pressure drop in the flow direction, u_D is the velocity of the fluid. It can be seen that the permeability is the ratio of the shear stress to the pressure drop for a system which obeys equation (4.3). For an anisotropic medium equation (4.3) becomes

$$\mathbf{u}_D = -\mu^{-1}\mathbf{K} \cdot \nabla P, \tag{4.4}$$

where $\mathbf{K}$ is a symmetric second-order tensor

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix}. \tag{4.5}$$

The permeability for simple porous structures such as packed spheres can be expressed in terms of the porosity and the particle diameter using the *hydraulic radius model* or the *Carman-Kozeny theory*. This is a semiheuristic model of flow through a porous medium which implements the concept of the hydraulic radius and *tortuosity*, τ, defined as the ratio of the distance to the displacement traveled by a fluid particle between two points.

The specific surface area of a porous matrix is defined by

$$A_0 = \frac{A_{fs}}{V_s}, \tag{4.6}$$

where A_{fs} is the interfacial area between the solid and fluid phases, and V_s is the volume of the solid. The hydraulic diameter is defined by

$$D_h = \frac{4 \times \text{void volume}}{\text{surface area}} = \frac{4\phi}{A_0(1-\phi)}. \tag{4.7}$$

The tortuosity is used to account for a correction of the pressure gradient,

$$\nabla p_{mod} = \frac{\nabla p}{\tau}, \tag{4.8}$$

and the pore velocity,

$$\mathbf{u}_D = \mathbf{u}_p \frac{\phi}{\tau} = -\frac{K}{\mu} \nabla p_{mod}. \tag{4.9}$$

The filter velocity is now determined by modifying the Hagen-Poiseulle equation [40],

$$u_p = -\frac{d^2}{32\mu} \frac{dp}{dx}, \tag{4.10}$$

for flow along straight tubes of diameter d. The modification of equation (4.9) involves the application of the tortuosity and the addition of a shape parameter k_0, and becomes

$$\mathbf{u}_p = -\frac{D_h^2}{16 k_0 \mu} \nabla p_{mod}, \tag{4.11}$$

where k_0=2 for a circular shape and 2-2.5 for rectangular, elliptical, and annular shapes. Combining equations (4.8), (4.9), and (4.11), we get,

$$K = \frac{\phi^3}{k_K (1-\phi)^2 A_0^2}, \tag{4.12}$$

where k_K is called the *Kozeny* constant. For a bed of uniformly-sized spherical particles of diameter d, we have,

$$K = \frac{\phi^3}{36 k_K (1-\phi)^2} d^2, \tag{4.13}$$

where $A_0 = 6/d$ and d is the diameter of the spherical particle with the same A_0. Taking the Kozeny constant to be 5 approximates the permeability in packed beds reasonably well and equation (4.13) becomes [42]

$$K = \frac{\phi^3}{180(1-\phi)^2} d^2. \tag{4.14}$$

The Darcy equation deviates from experimental results for liquid flows at high velocities and gas flows at very low and high velocities because it exhibits the characteristics of Stokes flow. At high velocities, the inertial effects dominate the viscous

effects. The existence of viscous and inertial asymptotes has been observed by Macdonald et al.[43]. Ergun[44], extended the hydraulic radius theory in order to account for the inertial effects in high-velocity flows. He developed a correlation based on the particle Reynolds number,

$$Re_d = \frac{\rho u_D d}{\mu}, \tag{4.15}$$

which was later modified by Macdonald et al. as,

$$\frac{-dp/dx}{\rho u_D^2} d \frac{\phi^3}{1-\phi} = \frac{180(1-\phi)}{Re_d} + 1.8, \tag{4.16}$$

for smooth surfaces, and

$$\frac{-dp/dx}{\rho u_D^2} d \frac{\phi^3}{1-\phi} = \frac{180(1-\phi)}{Re_d} + 4, \tag{4.17}$$

for rough surfaces. Multiplying both sides of equation (4.16) by $\rho u_D^2/d$ gives

$$-\frac{dp}{dx} = \frac{\mu}{K} u_D + 1.8 \frac{1-\phi}{\phi^3 d} \rho u_D^2. \tag{4.18}$$

This equation has been extended heuristically to multi-dimensional flows and becomes,

$$-\nabla p = \frac{\mu}{K} \mathbf{u}_D + 1.8 \frac{1-\phi}{\phi^3 d} \rho |\mathbf{u}_D| \mathbf{u}_D. \tag{4.19}$$

Equation (4.19) has not been verified experimentally and when simulating multi-dimensional flows, the coefficient of the second term on the right may need to be adjusted. A relatively large perpendicular velocity component causes a decrease in the coefficient.

4.3 The Semiheuristic Momentum Equations

As mentioned earlier, geometric considerations introduce a level of complexity which is currently difficult to overcome. As a result, it is very difficult to derive a momentum equation which would be of practical use. One possibility is to find extentions of Darcy's law to account for: 1) flow development, 2) boundary generated vorticity,

and 3) the high Reynolds number effect. In the first extension, the inertial and viscous effects were added with the assumption of steady flow, given as [40]

$$\frac{\rho_0}{\phi^2}(\mathbf{u}_D \cdot \nabla \mathbf{u}_D) = -\frac{\mu}{K}\mathbf{u}_D - \nabla p + \rho \mathbf{g} + \frac{\mu}{\phi}\nabla^2 \mathbf{u}_D. \tag{4.20}$$

There is no rigorous reasoning for the multiplication of the viscous term by $1/\phi$, and the inertial term by $1/\phi^2$. It is believed that since the pore velocities should be greater than the average values, then the viscous and inertial forces should be evaluated at the pore level. This is achieved with a multiplication of the velocity by the factor $1/\phi$. To account for high Reynolds number flows and gravitational effects, equation (4.20) was extended to,

$$\frac{\rho_0}{\phi^2}\left(\frac{\partial \mathbf{u}_{Di}}{\partial t} + \mathbf{u}_{Di} \cdot \nabla \mathbf{u}_{Di}\right) = -\frac{\mu}{K}u_{Di} - \frac{\partial p}{\partial x_i} + \rho \mathbf{g} + \frac{\mu}{\phi}\nabla^2 \mathbf{u}_{Di} - \frac{C_E}{K^{1/2}}\rho |u_{Di}| u_{Di}, \tag{4.21}$$

where C_E is the Ergun constant which represents the deviation from Stokes flow. A typical value for C_E is 0.55.

4.4 Packed Bed Model

A reduced form of equation (4.20) has been applied to packed beds by Cheng and Zhu[45], Vortmeyer and Schuster[46], and more recently Thomas and Faucher[29]. In their model, Thomas and Faucher also accounted for the heat transfer through the bed container. In the current model, the equations derived based on the work of Anderson and Jackson [22] are implemented. The following assumptions are made:

1) the analysis is performed using the method of volume-averaging,

2) the flow is assumed to be 2-d, steady, and fully developed,

3) only the z-component of the gas velocity is relevent,

4) the thermal model is 2-d, symmetric, and transient,

5) the porosity in the z-direction is constant,

6) heat generated by chemical reactions is small relative to that generated by the microwaves,

7) the electric field is assumed to be uniform, and

8) radiation heat transfer is neglected.

A more accurate model of the electric field could be obtained by solving Maxwell's equations via the finite element method, however this is beyond the scope of the current work. The transients are important in the thermal model because the dielectric properties can vary significantly with temperature. Thus, in order to predict the steady-state of the bed correctly, the thermal transients must be included [36]. The bed container is neglected because it requires a lot of computational effort and does not provide any additional information. The equation governing the fluid flow in the bed is given by

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\phi\mu\frac{\partial u_{Dz}}{\partial r}\right) = \frac{\partial P}{\partial z} + b\frac{1-\phi}{\phi^3 d_p}\rho u_{Dz}^2 + \frac{\mu}{K}u_{Dz}, \tag{4.22}$$

where b=1.75 is recommended [45] instead of 1.8 as presented in equation (4.18). Equation (4.14) is then substituted for K with a recommended value of 150 in the denominator rather than 180. The boundary conditions are

$$u_{Dz}(0) = 0, \tag{4.23}$$

and

$$\left.\frac{\partial u_{Dz}}{\partial r}\right|_0 = 0. \tag{4.24}$$

The equations governing the temperature distribution in the bed are given by [29]

$$(1-\phi)\rho C_{ps}\frac{\partial T_s}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r(1-\phi)k_s\frac{\partial T_s}{\partial r}\right) + (1-\phi)\frac{\partial}{\partial z}\left(k_s\frac{\partial T_s}{\partial z}\right) + h_v(T_f - T_s) + (1-\phi)\dot{q}, \tag{4.25}$$

and

$$\phi\rho C_{pf}\frac{\partial T_f}{\partial t} + \phi\rho C_{pf}u_{Dz}\frac{\partial T_f}{\partial z} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\phi k_f\frac{\partial T_f}{\partial r}\right) + \phi\frac{\partial}{\partial z}\left(k_f\frac{\partial T_f}{\partial z}\right) + h_v(T_s - T_f), \tag{4.26}$$

where T_s is the solid temperature, T_f is the gas temperature, and h_v is the volumetric heat transfer coefficient which is obtained from the correlation [10]

$$Nu = 2 + 1.2Re_d^{\frac{1}{2}}Pr^{\frac{1}{3}}. \tag{4.27}$$

The boundary conditions for the solid phase are

$$\frac{\partial T_s}{\partial r}\Big|_0 = 0, \quad -\phi_s k_s \frac{\partial T_s}{\partial r}\Big|_R = U(T_s - T_\infty)|_R, \tag{4.28}$$

and

$$\frac{\partial T_s}{\partial z}\Big|_0 = \frac{\partial T_s}{\partial z}\Big|_H = 0, \tag{4.29}$$

where R is the radius of the bed, H is the bed height, and U is the overall heat transfer coefficient given by [47]

$$U = \frac{1}{R_{tot} A} \tag{4.30}$$

where

$$R_{tot} = \frac{ln(r_o/r_i)}{k_g A} + \frac{1}{hA}. \tag{4.31}$$

In the above equation, k_g is the conductivity of the container, r_i is the radius of the bed, r_o is the radius of the bed plus the thickness of the container, and h is the heat transfer coefficient given by,

$$h = 0.29 \left(\frac{\Delta T}{L} \right). \tag{4.32}$$

For the gas phase,

$$\frac{\partial T_f}{\partial r}\Big|_0 = 0, \quad -\phi_f k_f \frac{\partial T_f}{\partial r}\Big|_R = U(T_f - T_\infty)|_R, \tag{4.33}$$

$$T_f(0) = T_{in}, \tag{4.34}$$

and

$$\frac{\partial T_f}{\partial z}\Big|_H = 0. \tag{4.35}$$

The radial porosity distribution is given by the equation [45],

$$\epsilon = \phi_0 \left(1 + C_1 exp \left[-\frac{N_1(R-r)}{d_p} \right] \right), \tag{4.36}$$

where ϕ_0 is the porosity at the center of the bed.

The proposed model was solved using the upwind finite-difference scheme following the procedure proposed by Patankar [48]. Equations (4.22) and (4.23) are solved iteratively. The u_{Dz}^2 term is approximated by $u_g u_{Dz}$ where u_g is the guessed value for u_{Dz}. The discretized form of the momentum equation for a general node is

$$-\left(\mu r \phi_f\right)_e \left(\frac{u_E - u_P}{\Delta r}\right) + \left(\mu r \phi_f\right)_w \left(\frac{u_P - u_W}{\Delta r}\right) + f_2 \left(\frac{r_e^2 - r_w^2}{2}\right) u_g u_P \qquad (4.37)$$
$$+ f_1 \left(\frac{r_e^2 - r_w^2}{2}\right) u_P = -\frac{\partial P}{\partial z} \left(\frac{r_e^2 - r_w^2}{2}\right),$$

where $\partial P/\partial z$ is constant because the flow is assumed to be fully developed. The subscripts E, W, N, and S represent the location of the immediate grid points which are located east, west, north, and south of the main grid point P respectively. The lower case subscripts e, w, n, and s represent the east, west, north, and south control volume faces located half-way between the grid points P and E, W, N, and S respectively. The superscripts 0, and 1 stand for the current and subsequent time steps respectively. The equation for the nodal velocities is obtained by rearanging equation (4.30)

$$\left[\frac{(\mu r \phi)_e}{\Delta r} + \frac{(\mu r \phi)_w}{\Delta r} + f_2 u_g \left(\frac{r_e^2 - r_w^2}{2}\right) + f_1 \left(\frac{r_e^2 - r_w^2}{2}\right)\right] u_P \qquad (4.38)$$
$$- \left[\frac{(\mu r \phi)_e}{\Delta r}\right] u_E - \left[\frac{(\mu r \phi)_w}{\Delta r}\right] u_W = -\frac{\partial P}{\partial z} \left(\frac{r_e^2 - r_w^2}{2}\right),$$

where the functions f_1, and f_2 are given by

$$f_1 = \frac{150 \mu_f (1-\phi)^2}{d_p^2 \phi^3}, \qquad (4.39)$$

and

$$f_2 = \frac{1.75 \rho_f (1-\phi)}{d_p \phi^3}. \qquad (4.40)$$

The discretized form of the energy equation with the upwind scheme for a general

node is

$$(\rho_C p\phi)_i \Delta z \left(\frac{r_e^2 - r_w^2}{2}\right)\left(\frac{T_{Pi}^1 - T_{Pi}^0}{\Delta t}\right) = (rk_i\phi_i)_e \Delta z \left(\frac{T_{Ei}^1 - T_{Pi}^1}{\Delta r}\right) \tag{4.41}$$
$$-(rk_i\phi_i)_w \Delta z \left(\frac{T_{Pi}^1 - T_{Wi}^1}{\Delta r}\right) + (k_i\phi_i)_n \left(\frac{r_e^2 - r_w^2}{2}\right)\left(\frac{T_{Ni}^1 - T_{Pi}^1}{\Delta z}\right)$$
$$-(k_i\phi_i)_s \left(\frac{r_e^2 - r_w^2}{2}\right)\left(\frac{T_{Pi}^1 - T_{Si}^1}{\Delta z}\right) - (\phi_i\rho_i C p_i u_{Dzi})_n \left(\frac{r_e^2 - r_w^2}{2}\right) T_{Pi}^1$$
$$+(\phi_i\rho_i C p_i u_{Dzi})_s \left(\frac{r_e^2 - r_w^2}{2}\right) T_{Si}^1 + h_v \left(\frac{r_e^2 - r_w^2}{2}\right) \Delta z (T_{Pi}^1 - T_{Pj}^1) + \phi_i \dot{q}_i$$

where the subscripts i and j represent either the solid or fluid phase and j is not i. In the above equation, u_{Dzs} and $\dot{q}_f$ are zero. Rearanging equation (4.34) in terms of the nodal temperatures gives

$$[a_t + a_E + a_W + a_N + a_S + a_h] T_{Pi}^1 - a_E T_{Ei}^1 - a_W T_{Wi}^1 - a_N T_{Ni}^1 \tag{4.42}$$
$$-a_S T_{Si}^1 - a_h T_{Pj}^1 = a_t T_{Pi}^0 + \epsilon_i \dot{q},$$

where

$$a_t = \frac{(\rho_C p\phi)_i}{\Delta t} \Delta z \left(\frac{r_e^2 - r_w^2}{2}\right), \tag{4.43}$$

$$a_E = \frac{(rk_i\phi_i)_e \Delta z}{\Delta r}, \tag{4.44}$$

$$a_E = \frac{(rk_i\phi_i)_w \Delta z}{\Delta r}, \tag{4.45}$$

$$a_N = \frac{(k_i\phi_i)_n}{\Delta z} \left(\frac{r_e^2 - r_w^2}{2}\right), \tag{4.46}$$

$$a_S = \frac{(k_i\phi_i)_s}{\Delta z} \left(\frac{r_e^2 - r_w^2}{2}\right) + (\phi_i\rho_i C p_i u_{Dzi})_n \left(\frac{r_e^2 - r_w^2}{2}\right), \tag{4.47}$$

$$a_h = h_v \left(\frac{r_e^2 - r_w^2}{2}\right) \Delta z. \tag{4.48}$$

Chapter 5

Results for the Packed Bed

In this study, air flow through a packed bed consisting of gamma alumina particles is examined. The effects of the particle diameter and inlet velocity on the bed temperature and the temperature difference between both phases are analyzed. In the first study, these effects are observed for particle diameters of 0.5, 1, 2, and 3 mm. The gas enters the bed at 10 cm/s and at 300.15 K. The radius and height of the bed are 1 cm and 2.54 cm respectively as shown in Figure 1 of chapter 1. These dimensions were chosen as typical of experimental arrangements reported in the literature. For computational purposes, the bed was subdivided into 30 radial cells and 20 axial cells with a time step of 100 seconds. This was determined after computing the percent error of the velocity and temperature profiles using the 30 by 20 grid relative to those using a 40 by 40 grid. The largest errors occured for the 0.5mm diameter case which were 3 and 4.2 percent for the velocity and temperature profiles respectively. The computations were performed with a uniform electric field of 5 kV/m in all cases.

Figures 5.1-5.4 show the time evolution of the radial and axial temperature profiles of the gas and solids for the 3 mm particles. Note that time increases towards the left. The points for the radial profiles are located midway along the bed height and the points for the axial profiles are located midway between the center and the wall of the bed. It can be seen that the radial and axial temperatures increase in time. The smaller particles produce the greatest temperatures because their arrangement yields the smallest interstices, thus creating more solid contact. Equation (4.29)

captures this effect by yielding smaller porosity values within the bed for smaller particle diameters.

Figures 5.5-5.8 show the steady-state temperature profiles for the gas and solid for the 3 mm particles. The steady-state temperature profiles reveal that radial temperature gradients are relatively small and are greater at the inlet of the bed. The small temperature gradient can be attributed to the uniform electric field assumption. The temperature gradients are greater at the inlet because the gas velocity is higher towards the wall and the gas is cooler at the inlet of the bed. As the gas propagates through the bed it absorbs heat from the hotter particles and its temperature increases with bed height.

Figures 5.9-5.12 display the temperature difference between the gas and solid and the gas velocity profiles for the particle diameters discussed above. The gas velocity is larger in a small region close to the wall than in the rest of the bed because the larger porosity in that region produces channeling of the gas. Also, the gas velocity decreases with increasing particle diameter because the smaller interstices resulting from the smaller particles cause larger pressure drops at the same inlet velocity. The temperature difference between the solid and gas increases with increasing particle diameter because the larger interstices do not allow the gas to remove heat as efficiently from the solids as the smaller interstices. Since the porosity is larger near the wall, the temperature difference between the solid and gas remains larger for a greater distance along the bed height. The variation of the temperature difference with particle diameter is shown in figure 13 for two locations in the bed. The locations of the points were selected in low and high velocity regions close to the bed inlet, that is near the bed center and wall respectively. The node near the wall shows a larger temperature difference than the one near the bed center because the porosity near the wall is higher and hence the fluid offers more resistance to heating.

In the next study, we examine the effects of inlet velocity on the parameters discussed above. From the above study, we see that a temperature difference occurs throughout more of the bed when the 3 mm particles are used and hence the effects of the inlet gas velocity are investigated for these particles. From the above study,

we have data for an inlet velocity of 0.1 m/s. Additional data for inlet velocities of 0.05, 0.2, and 0.4 m/s was also obtained.

The time evolution of the radial and axial temperature profiles for the gas and 3 mm solid particles at the inlet velocities mentioned above are similar to those in figure 1 and will not be shown. Again all temperatures increase with time as expected for the same reasons dicussed above. The temperatures are highest for the smallest inlet gas velocity (0.05 m/s) because the gas does not remove heat as efficiently from the solids as it does at higher gas velocities. Hence, the temperatures decrease with increasing inlet gas velocity. The steady-state temperature distributions are similar to those in figures 5.5-5.8.

Figures 5.14-5.16 exhibit the temperature difference and the gas velocity profiles for the 3 mm particles at the inlet gas velocities discussed above. The gas velocity distributions are similar to those in the previous study and the gas velocity in the bed is directly proportional to the inlet gas velocity. The temperature difference between the solid and gas decreases at the bed inlet with increasing inlet velocity because heat is removed more efficiently from the solids. However, since fluid at a higher velocity does not reside in the container for as long as a lower velocity fluid, the temperature difference penetrates further into the bed for the higher velocity fluid. Figure 5.17 shows the variation of the temperature difference with inlet gas velocity for the two points from the previous study. It can be seen that the temperature difference decreases with increasing inlet gas velocity.

In our final study, we investigate the effects of the electric field on the temperature difference in the bed. For this study, we assume a bed composed of 3mm particles with an inlet gas velocity of 0.4 m/s because the previous study revealed that the temperature difference penetrates further into the bed for these values. We already analyzed the case for an electric field of 5 kV/m. We now investigate the cases for electric fields of 10, 15, and 20 kV/m.

Figures 5.18-5.20 show the temperature difference and velocity profiles for the electric fields above. It is seen from these figures that an increase in the magnitude of the electric field induces an increase in the temperature difference. This

occurs because the heat generated in the solid is directly proportional to square of the magnitude of the electric field. Figure 5.21 shows the variation of the temperature difference with the electric field for the same two points as in the previous studies. The temperature difference increases with an increase in electric field as expected.

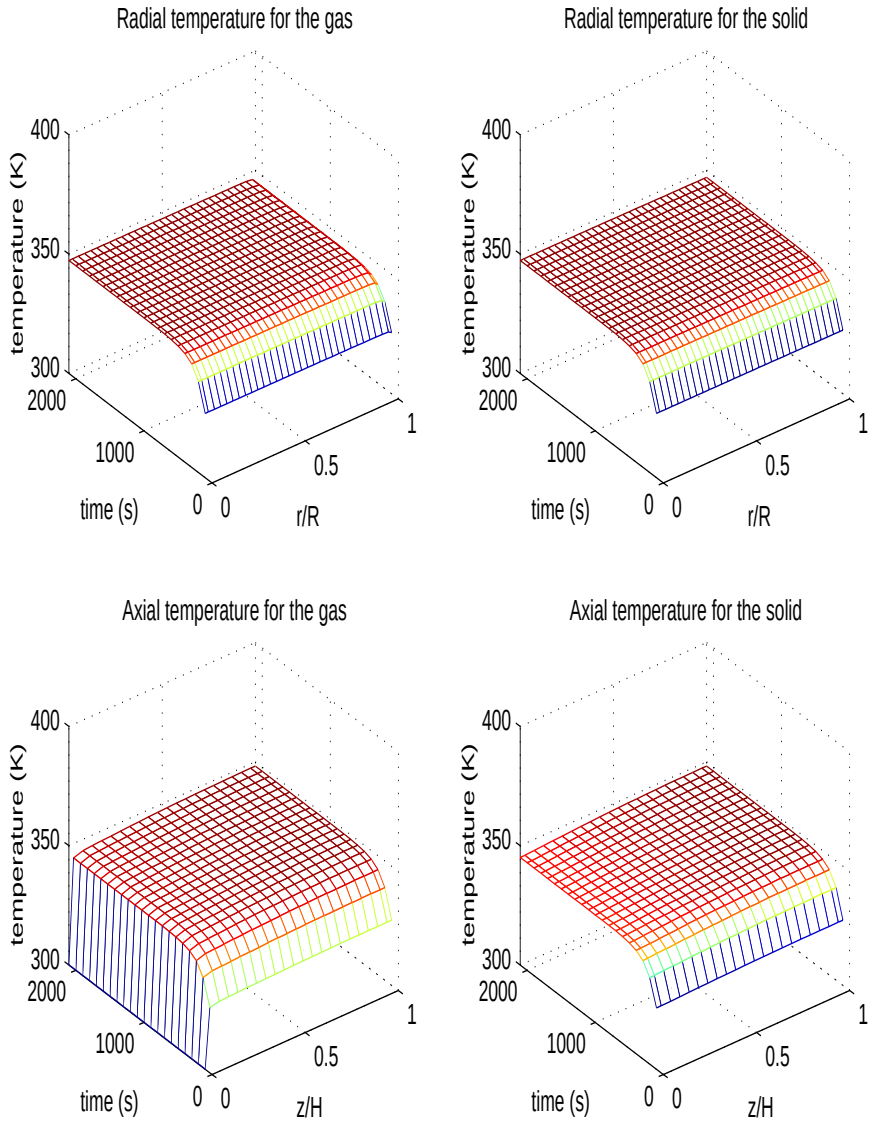


Figure 5.1: Temporal radial and axial temperature profiles for gas and 0.5 mm solid particles.

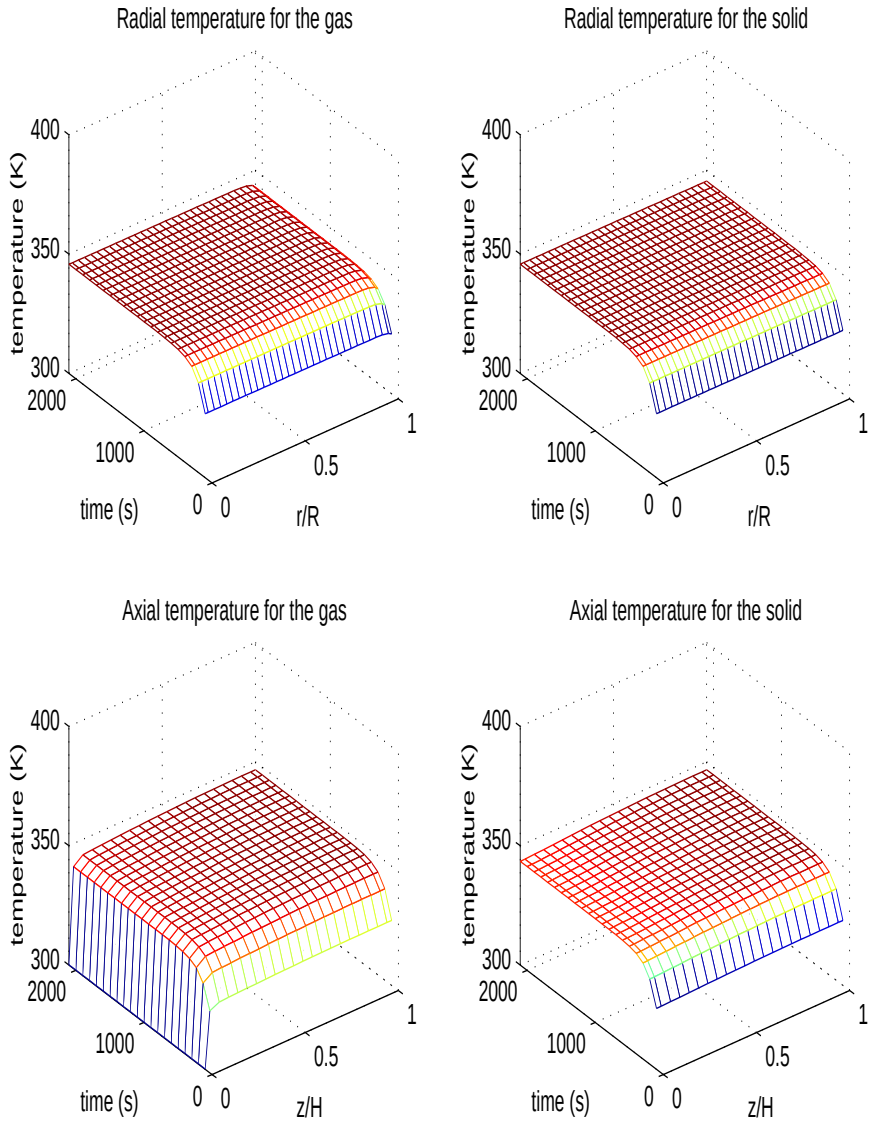


Figure 5.2: Temporal radial and axial temperature profiles for gas and 1 mm solid particles.

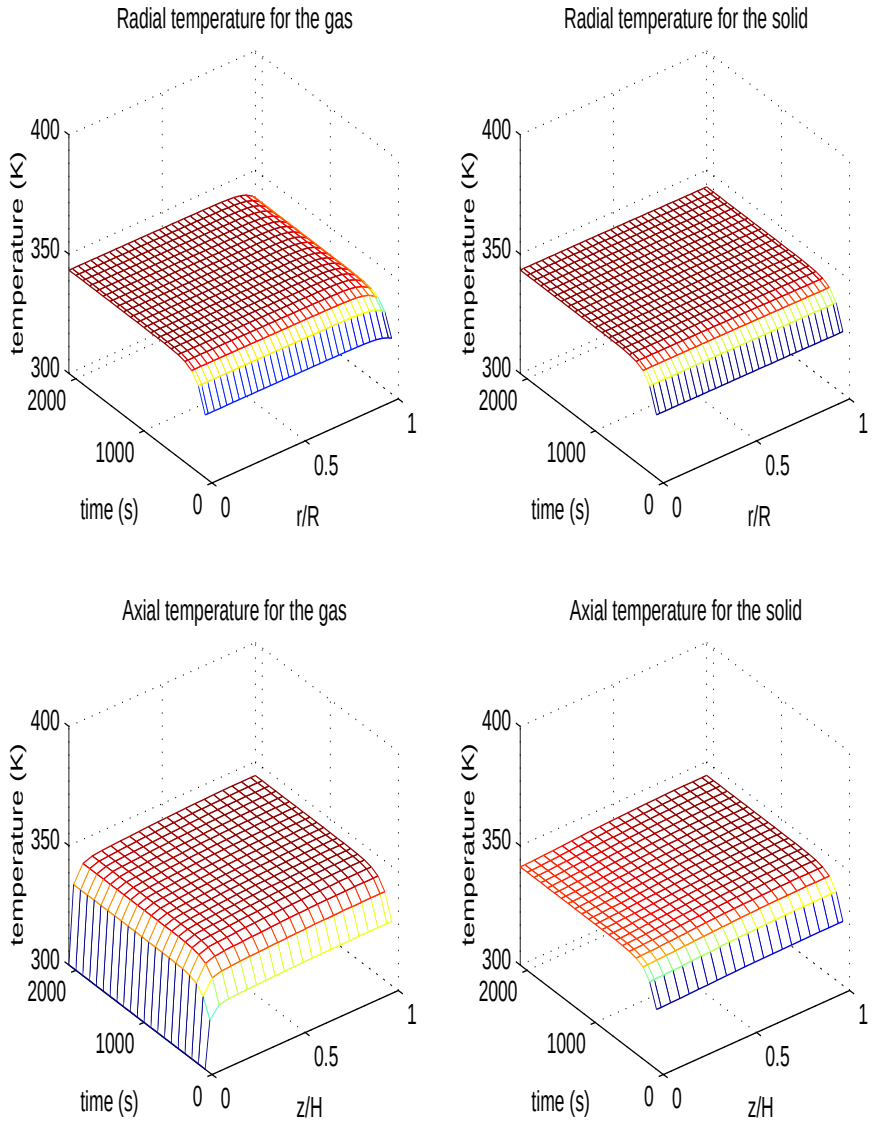


Figure 5.3: Temporal radial and axial temperature profiles for gas and 2 mm solid particles.

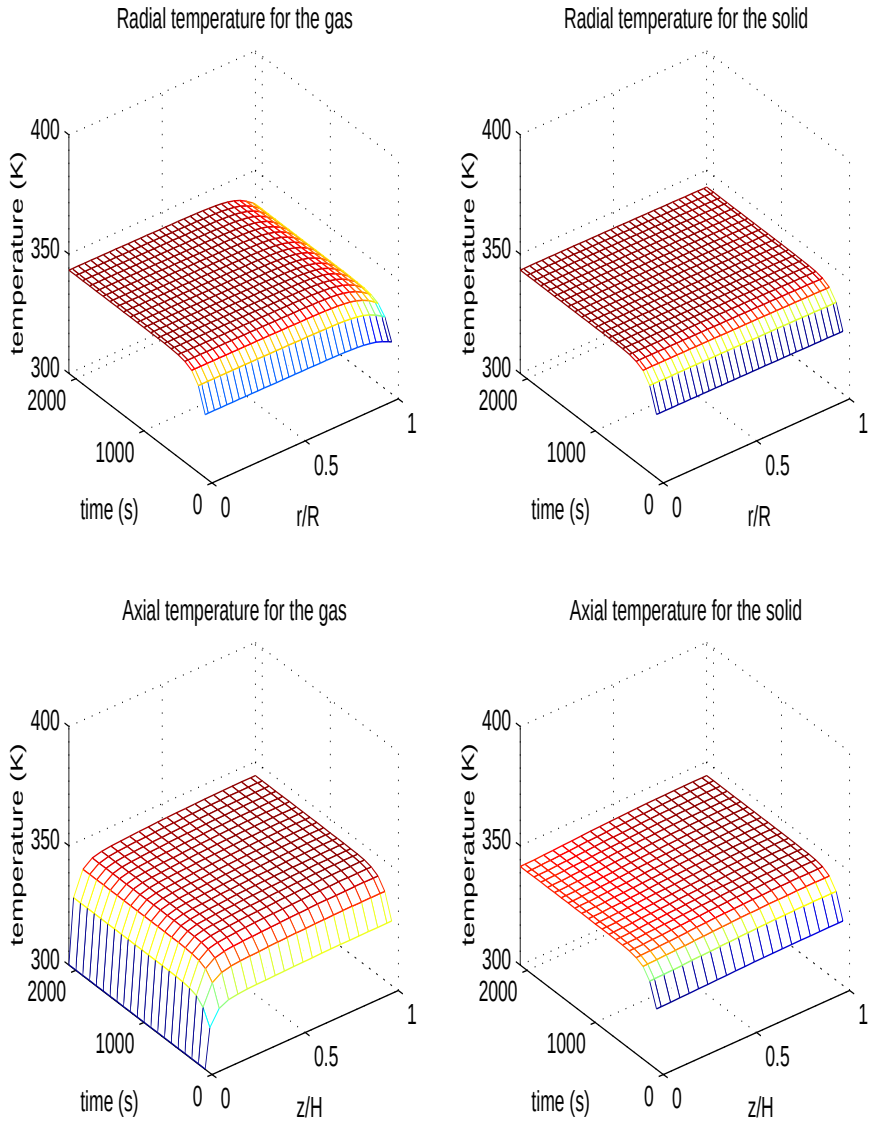


Figure 5.4: Temporal radial and axial temperature profiles for gas and 3 mm solid particles.

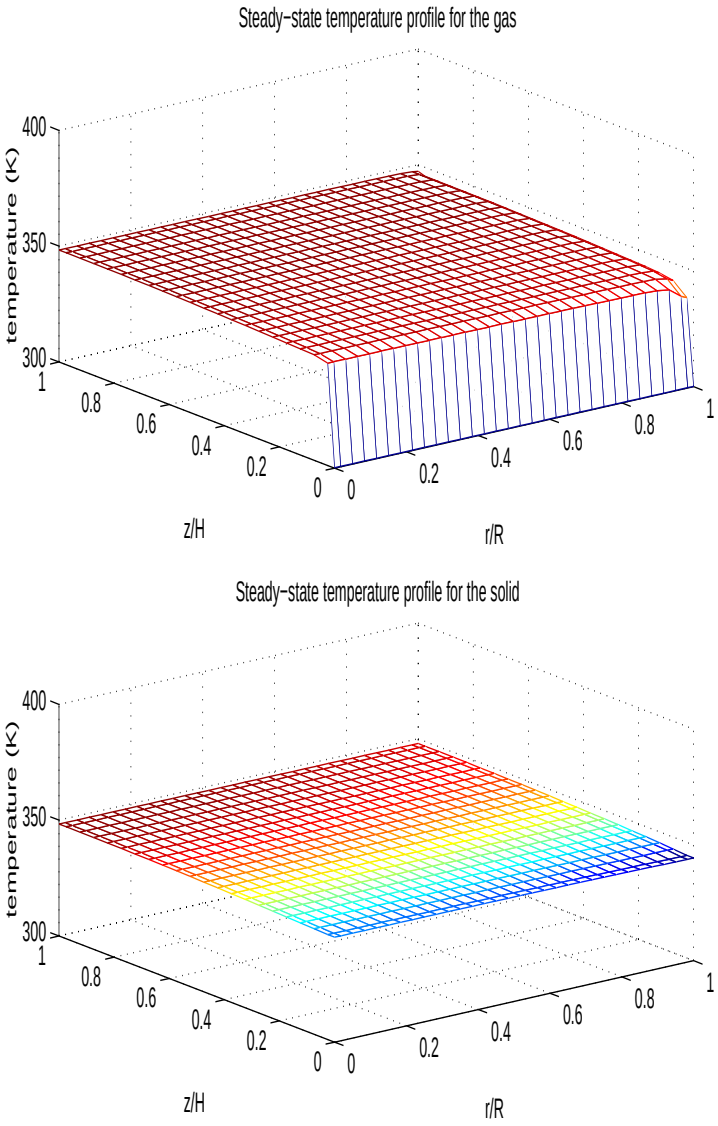


Figure 5.5: Steady-state temperature profiles for gas and 0.5 mm solid particles.

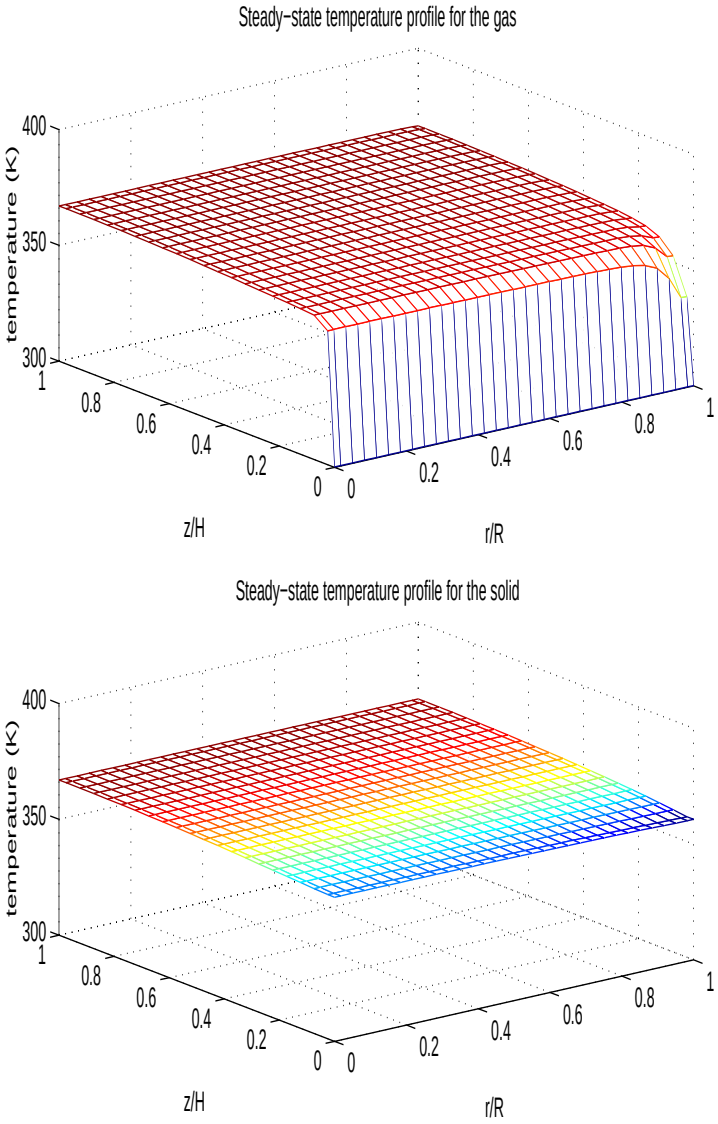


Figure 5.6: Steady-state temperature profiles for gas and 1 mm solid particles.

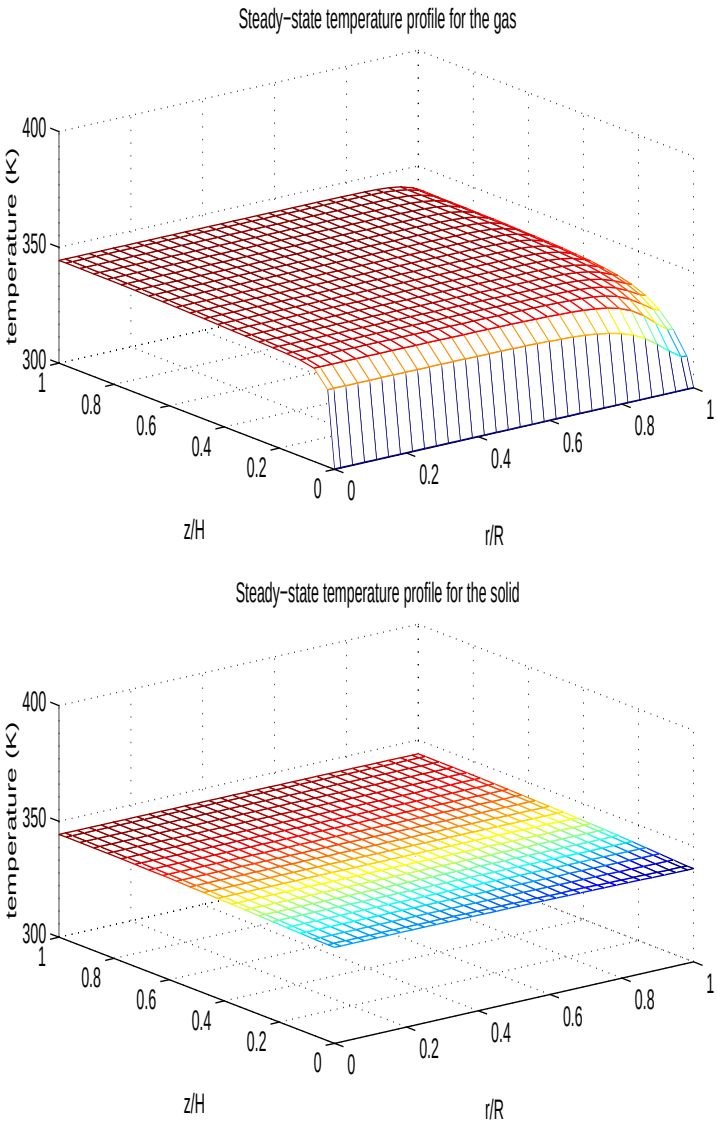


Figure 5.7: Steady-state temperature profiles for gas and 2 mm solid particles.

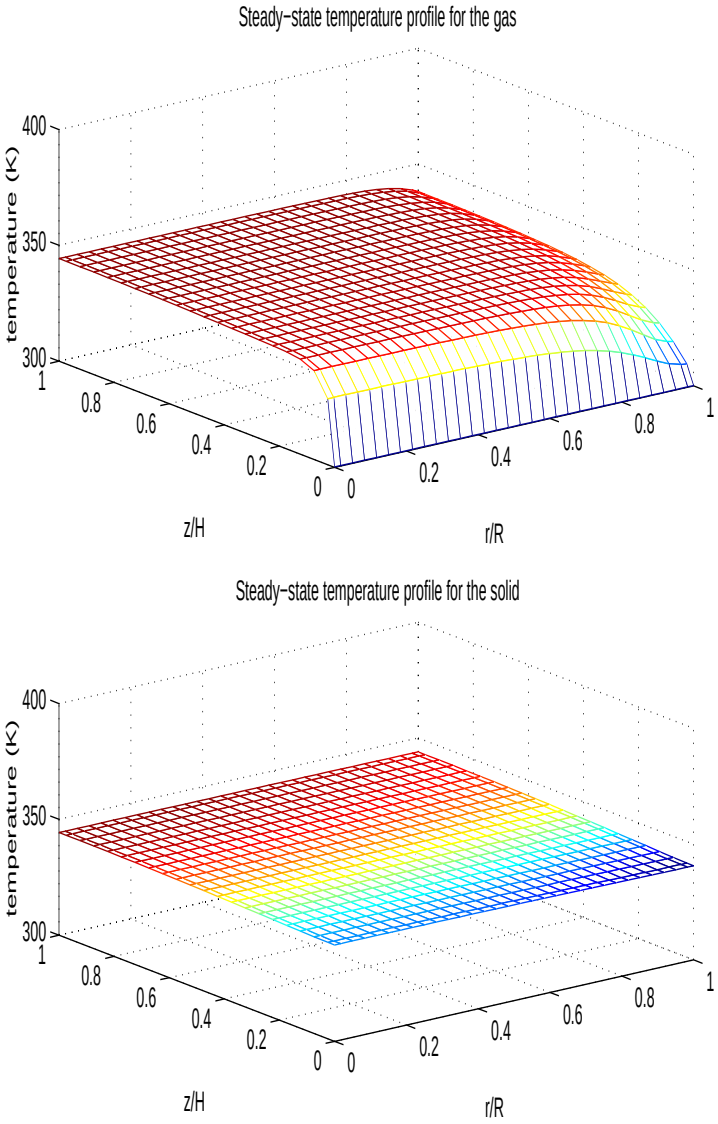


Figure 5.8: Steady-state temperature profiles for gas and 3 mm solid particles.

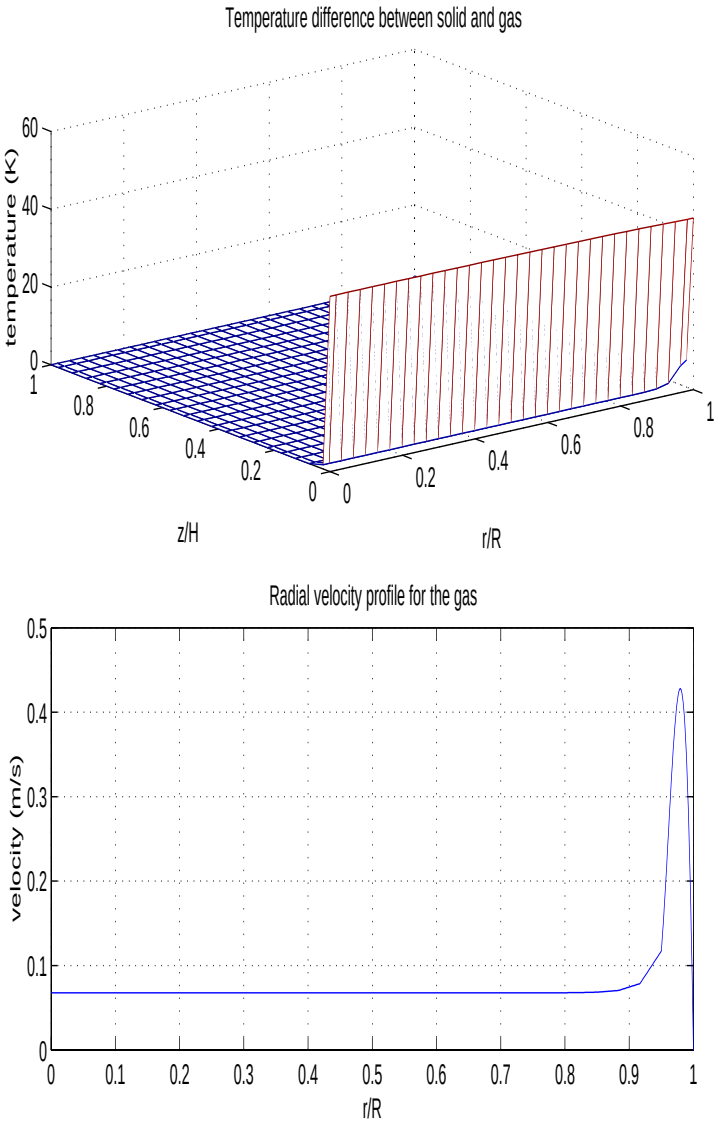


Figure 5.9: Temperature difference and gas velocity profile for 0.5 mm solid particles.

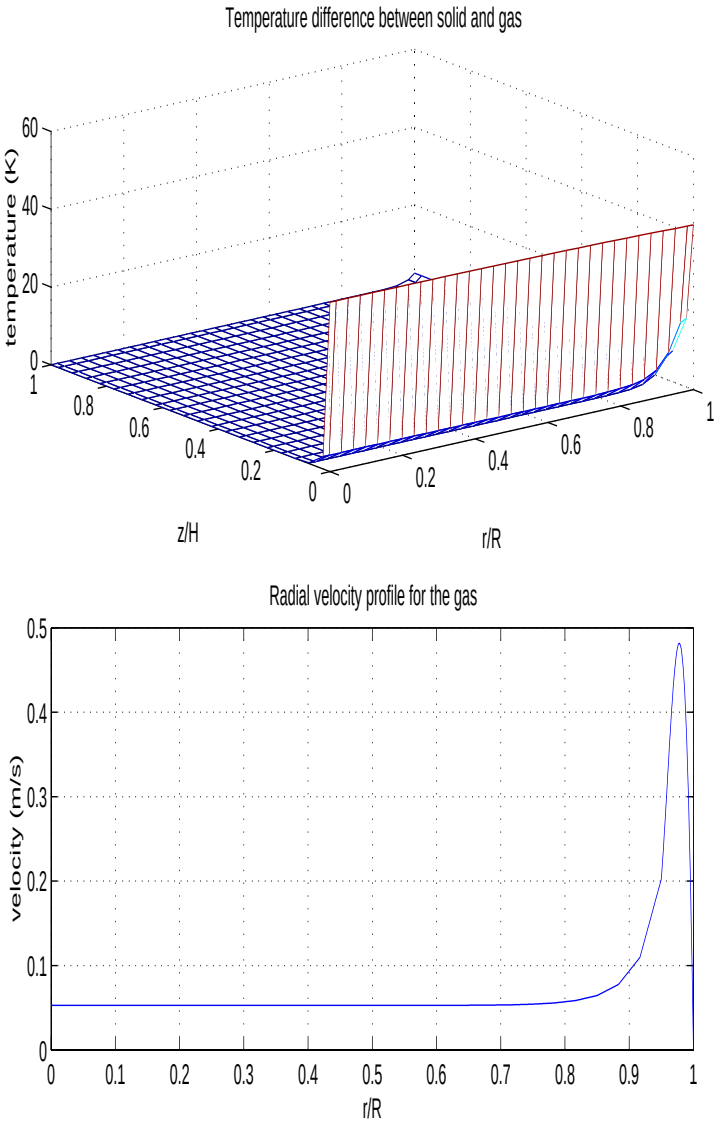


Figure 5.10: Temperature difference and gas velocity profile for 1 mm solid particles.

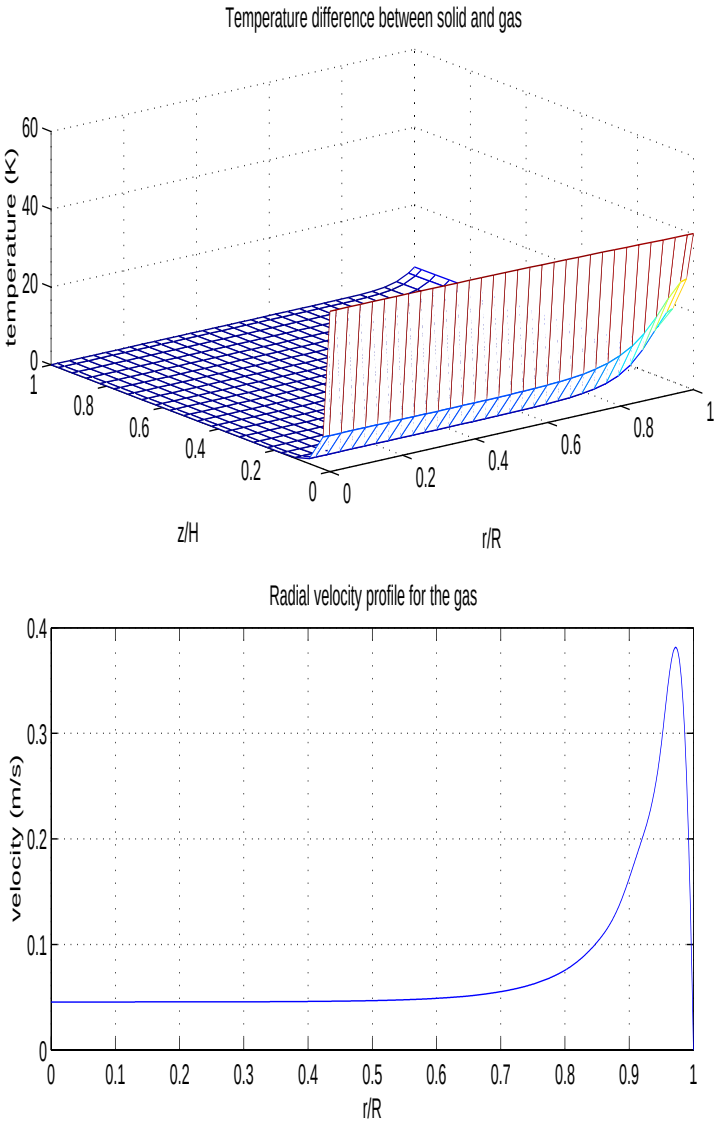


Figure 5.11: Temperature difference and gas velocity profile for 2 mm solid particles.

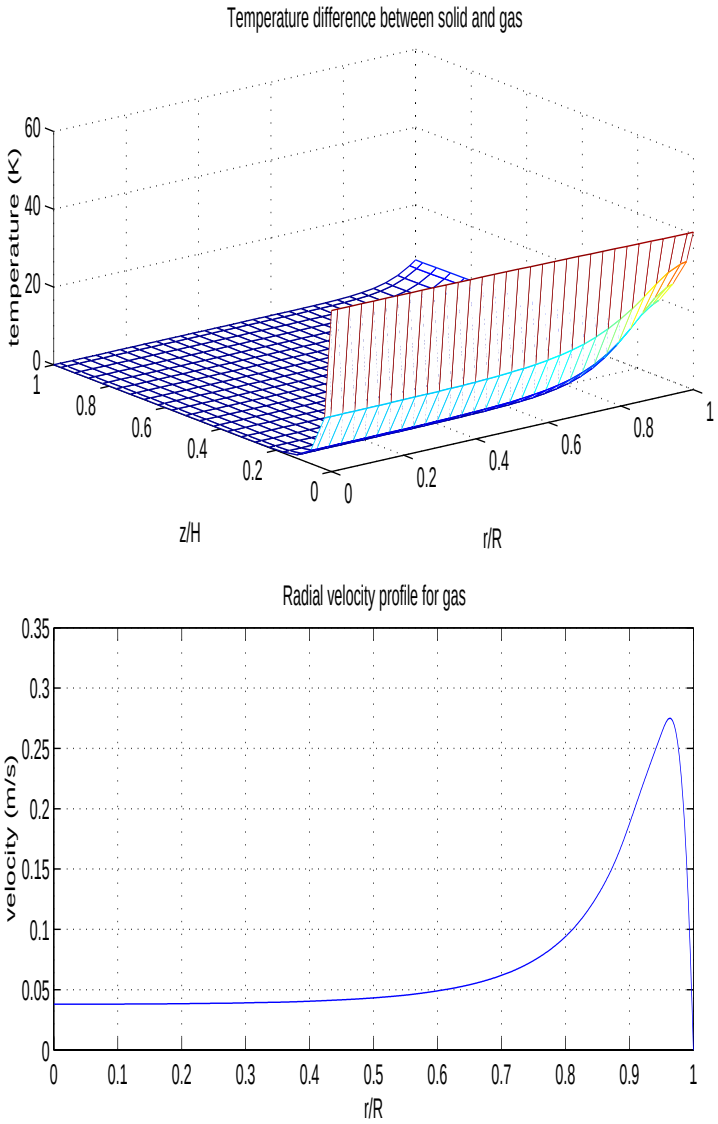


Figure 5.12: Temperature difference and gas velocity profile for 3 mm solid particles.

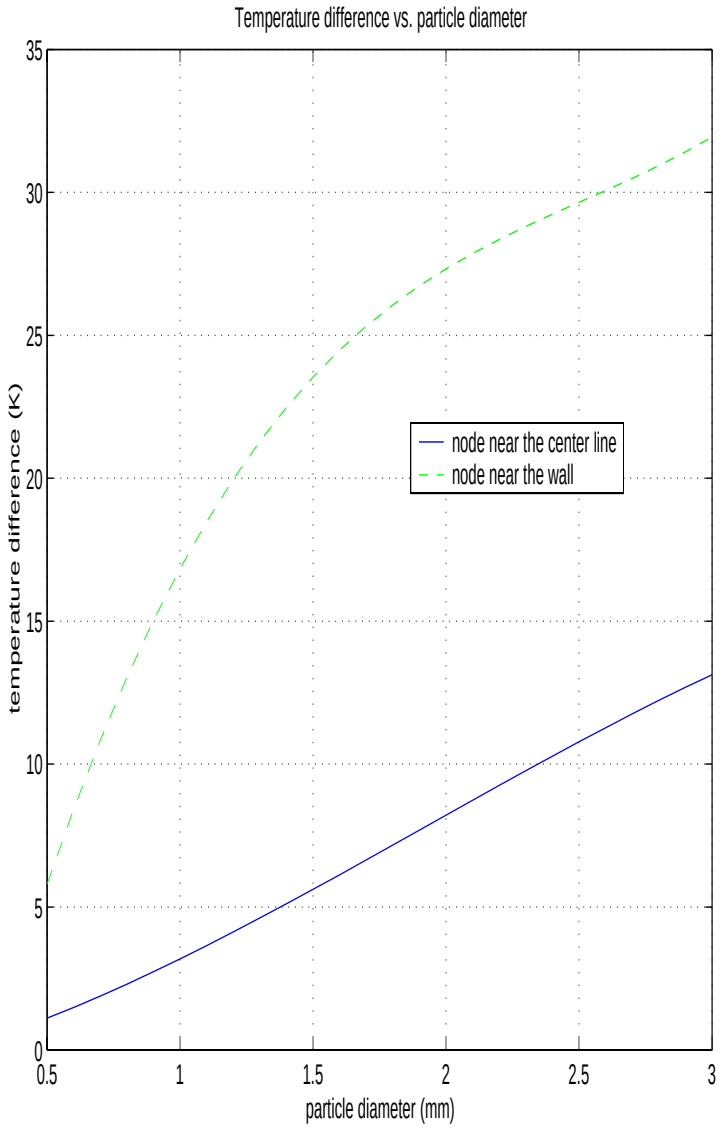


Figure 5.13: Effect of particle diameter on the temperature difference between the gas and solid.

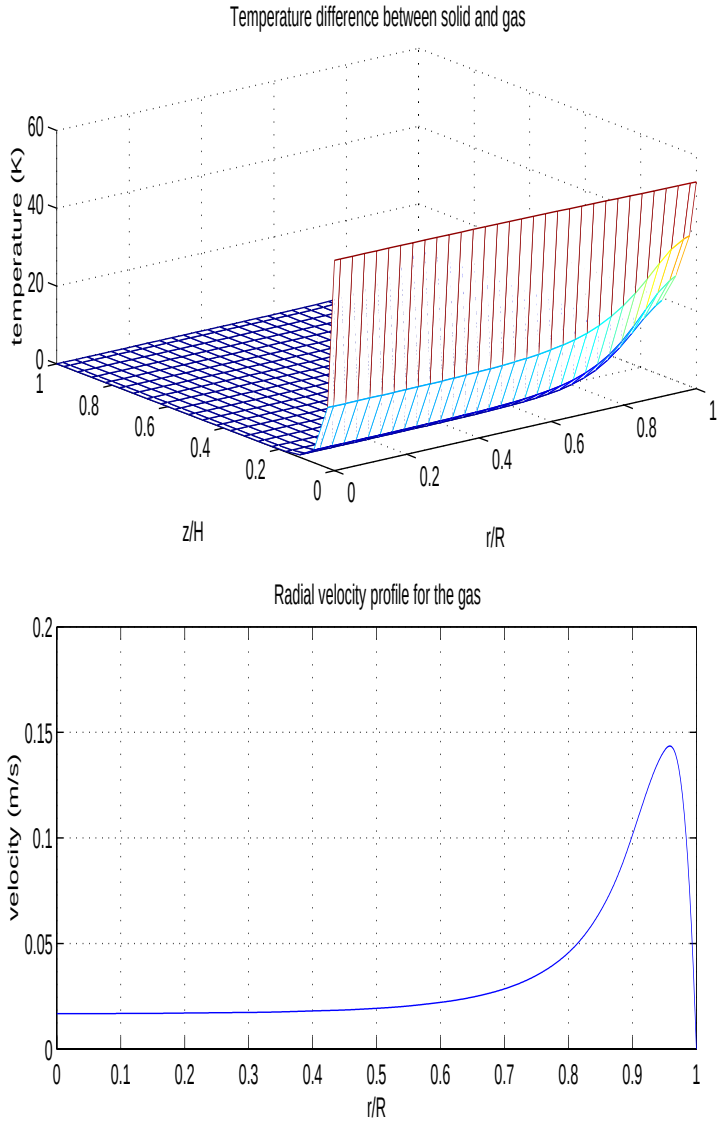


Figure 5.14: Temperature difference and gas velocity profile at an inlet velocity of 0.05 m/s.

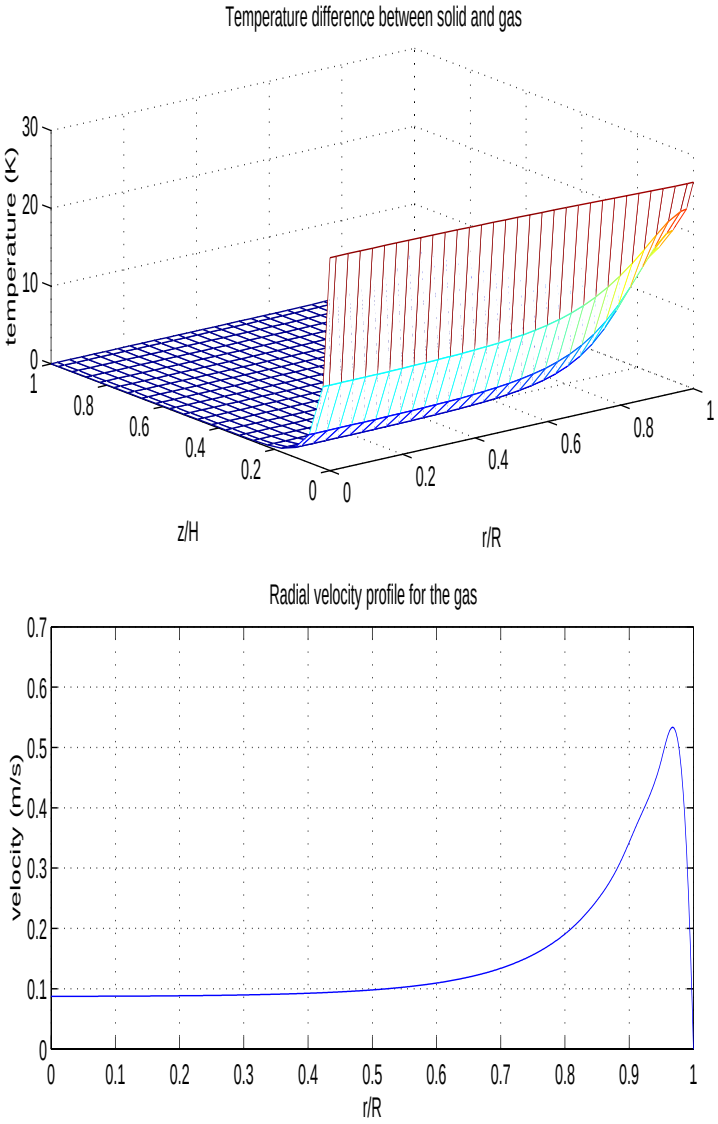


Figure 5.15: Temperature difference and gas velocity profile at an inlet velocity of 0.2 m/s.

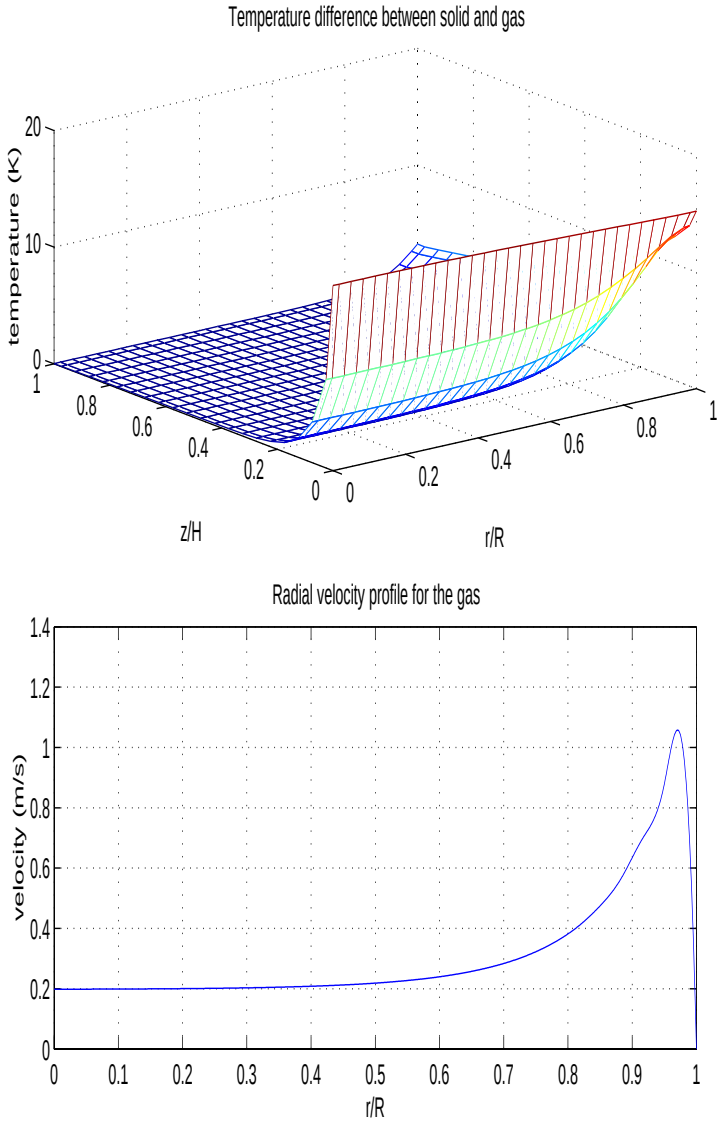


Figure 5.16: Temperature difference and gas velocity profile at an inlet velocity of 0.4 m/s.

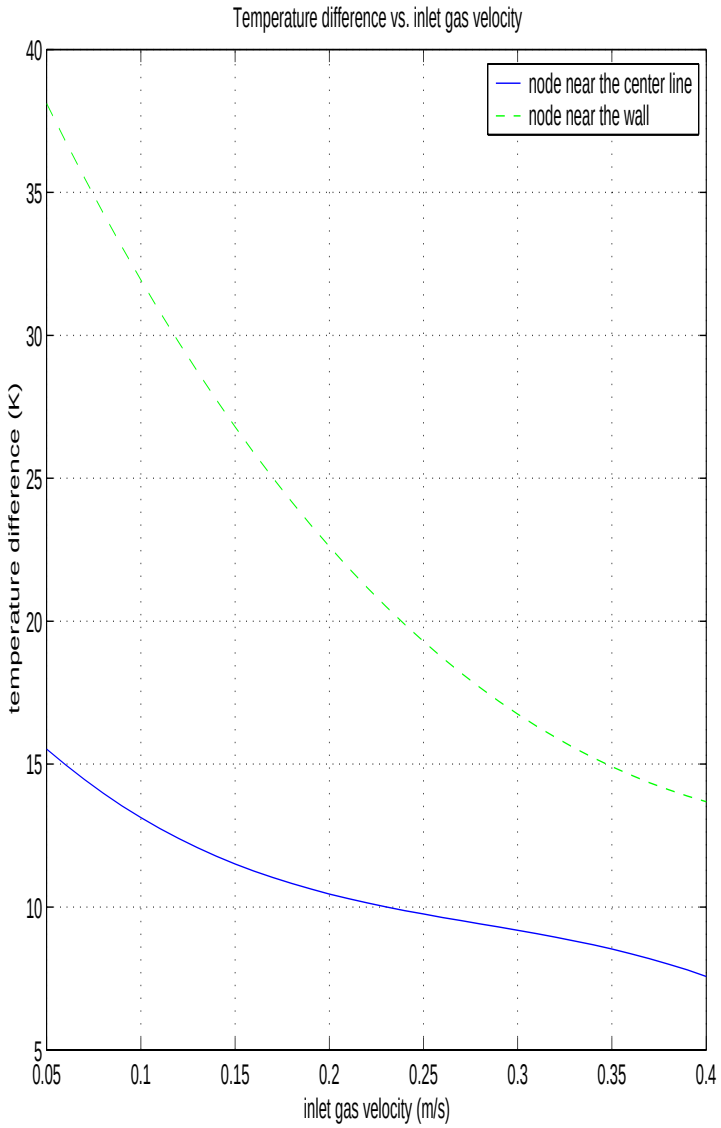


Figure 5.17: Effect of inlet gas velocity on the temperature difference between the gas and solid.

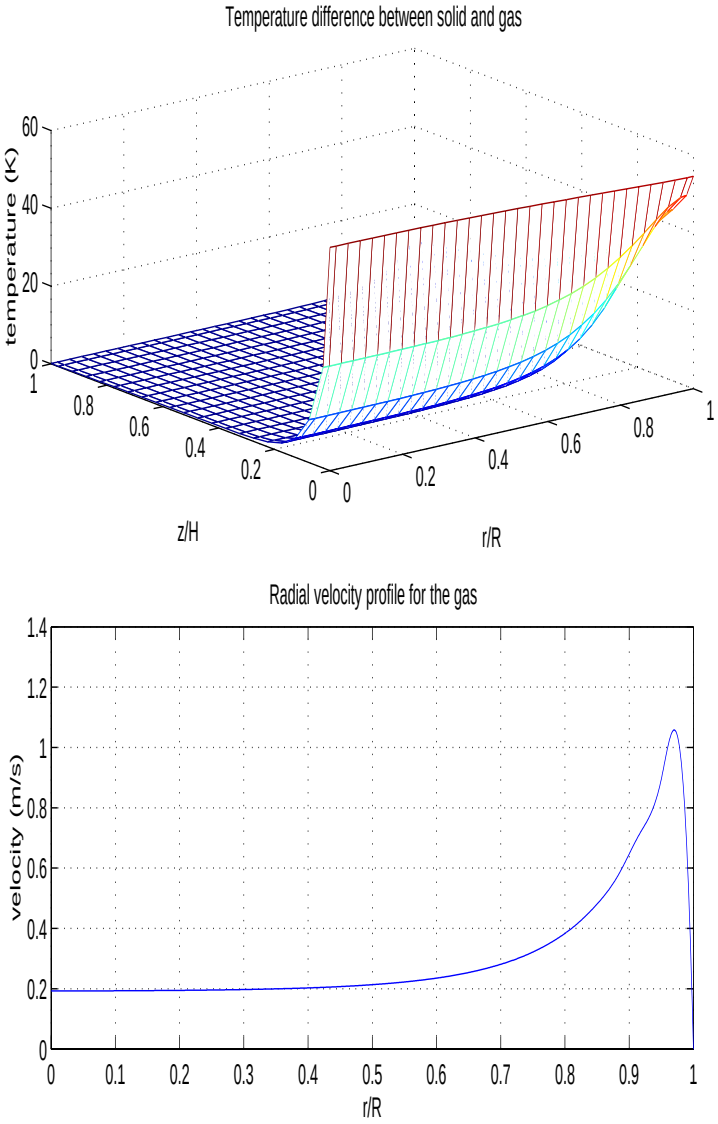


Figure 5.18: Temperature difference and gas velocity profile for an electric field of 10 kV/m.

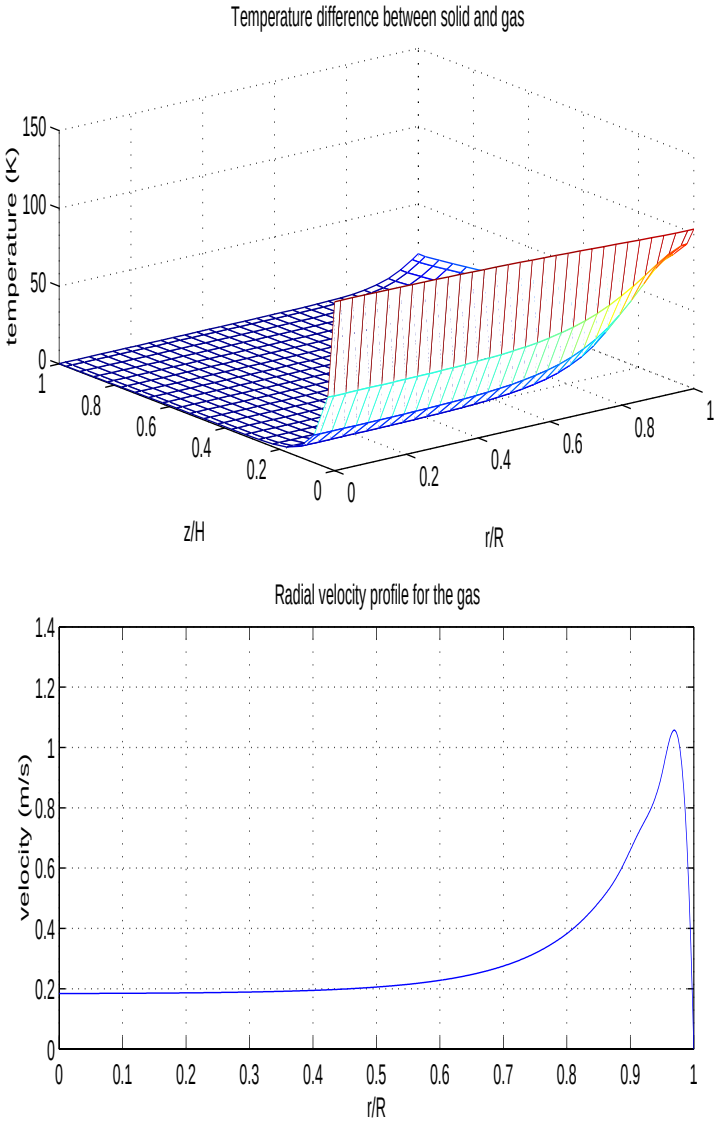


Figure 5.19: Temperature difference and gas velocity profile for an electric field of 15 kV/m.

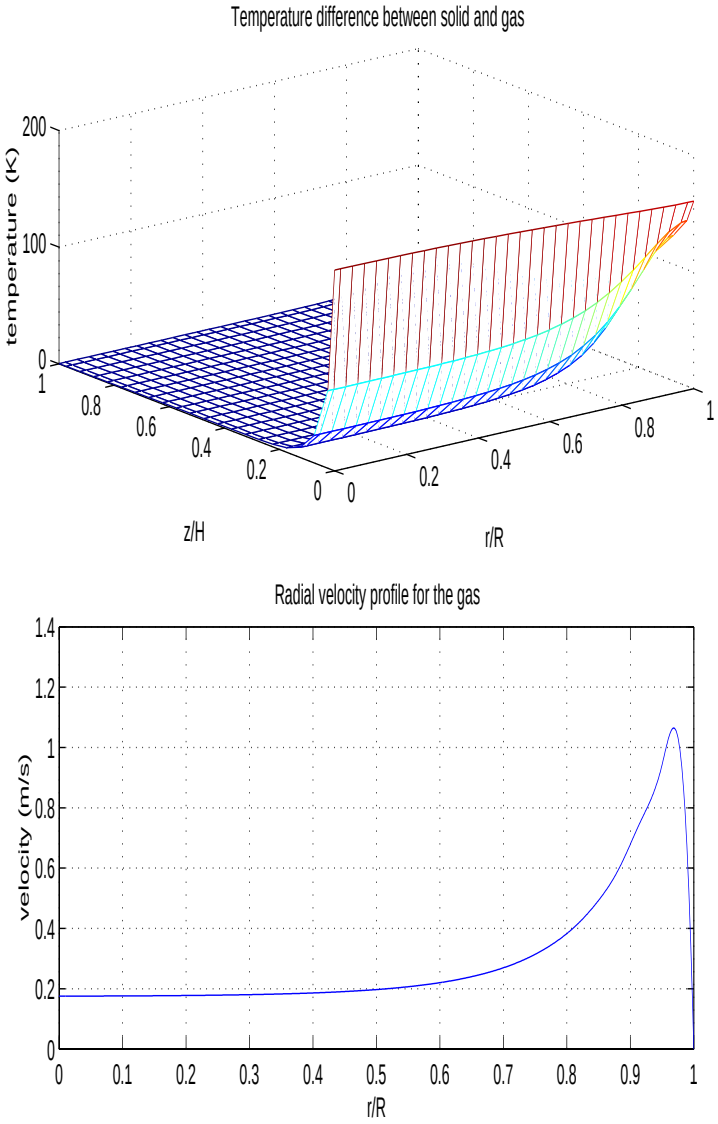


Figure 5.20: Temperature difference and gas velocity profile for an electric field of 20 kV/m.

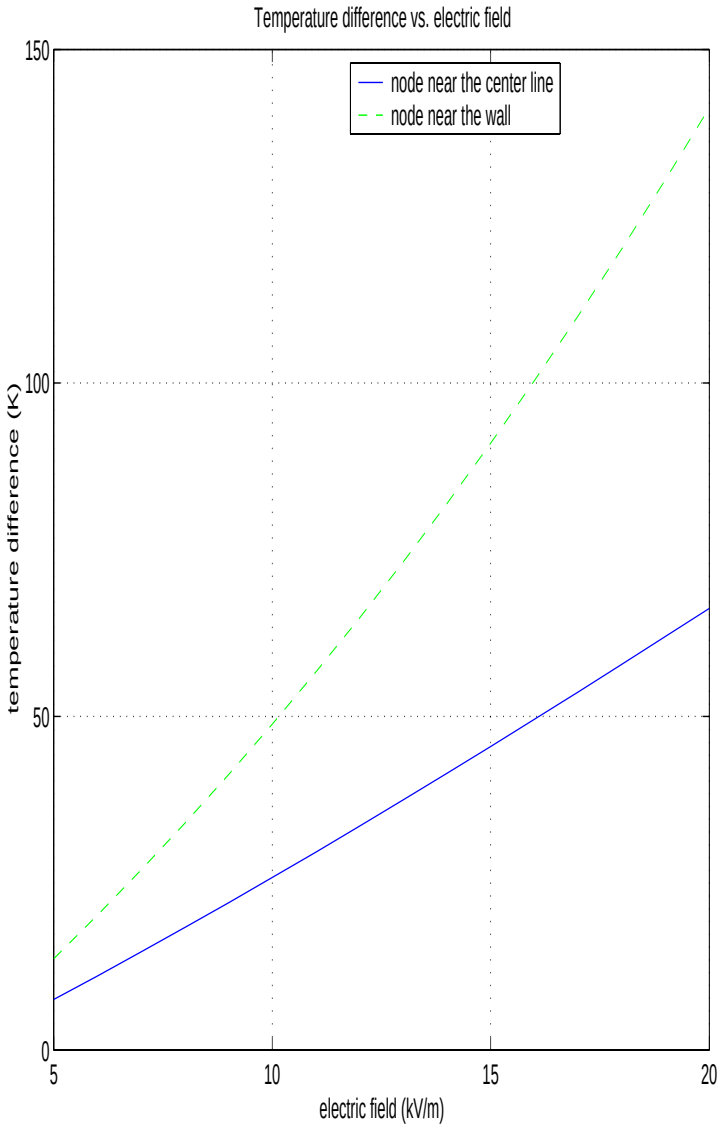


Figure 5.21: Effect of electric field on the temperature difference between the gas and solid.

Chapter 6

Fluidized Beds

6.1 Fluidization

Fluidization is the process by which a group of particles is suspended by a fluid flowing through a confined region. This occurs when the drag force exerted by the fluid on the particles exceeds the total weight of the particles. Some areas in which fluidized beds are ultilized include solids drying, chemical reactors and combustors [1].

The main components of a fluidized bed excluding the external equipment are the gas distributor which is placed at the bottom of the bed and the surrounding container. The purpose of the distributer is to control the distribution of the inlet gas. The distributor design is important in producing high quality fluidization. Some common distributors include ceramic or sintered metal porous plate distributors and perforated plate distributors. A ceramic or sintered metal porous plate distributor is most commonly used in small-scale studies because of its ability to deliver the gas uniformly across the bed inlet. Although the gas distribution is not perfectly uniform at the inlet, it is sufficiently close. These distributors have several disadvantages: 1) high pumping power is required due to large pressure-drops, 2)high cost for materials, 3) clogging caused by corrosion or fine particles. However, compacted wire plates or sandwiched beds of small particles are used despite the disadvantages. Perforated plate distributors are frequently used in industry because they are cost-efficient and

[1]Most of the discussion in this section is based on Ref. [49].

easy to fabricate. These distributors are not used in a high-temperature or a highly reactive environment with large loads because they tend to undergo severe deflections which can produce undesirable results.

Fluidization occurs in several states. When the fluid is passed through the bed at flow rates that cannot induce a drag force sufficient to overcome the weight of the bed, the fluid percolates through the voids and the bed remains fixed. If the flow rate is increased to the point at which the drag force just balances the total weight of the particles, the pressure drop in any portion of the bed equals the weight of the fluid and particles in that region. The inlet gas velocity which causes this to occur is called the minimum fluidization velocity, u_{mf}, and a bed in this state is said to be at *minimum fluidization* and has a uniform particle distribution. Such beds are called *homogeneous*. When the flow rate is incresed beyond minimum fluidization, instabilities are accompanied by bubbling and channeling of gas. Such beds are refered to as *bubbling* or *heterogeneous* fluidized beds. The volume in these beds does not differ by much compared with the volume at minimum fluidization. The size of the bubbles increase with the height of the bed. For sufficiently deep beds, the bubble may become as large as the length of the bed. This phenomenon is known as *slugging*. When slugging occurs in beds of fine particles, the particles flow downward by the wall around the rising bubble. For beds made up of coarse particles, the particles above the bubble are displaced upwards in a manner which is similar to a piston-cylinder configuration. The particles then fall down and the slug vanishes. Another slug is formed at about this time and an oscillatory pattern occurs. When the terminal velocity of the particles is exceeded by the gas, the particles propagate in turbulent clusters due to entrainment and the size and shape of the gas voids varies. A bed in this state is called a *turbulent fluidized bed*. A further increase in gas flow rates results in the exiting of the solid particles from the bed. Eventually the solids concentration is very low and the bed is said to be a *dilute-* or *lean-phase* fluidized bed. A fluidized bed is said to be *dense* if there is a distinct upper surface. Within a dense bed, the solids concentration is at least 25 percent. The primary focus of this work is on dense and homogeneous gas-solid fluidized beds. The homomgeneous fluidization state will

be discussed in more detail in the next section.

The quality of the fluidization process depends strongly on the particle size and the gas to solid density ratio. Other factors which may influence the quality of fluidization include vessel geometry, distributor type, and type of solids used. Beds consisting of fine particles need to be agitated to maintain quality fluidization because the particles tend to clump together when they become moist. This is accomplished either with a mechanical stirrer or by using the kinetic energy of high flow rate gas jets to agitate the solids. Fine particle beds can be large and deep because these beds can be fluidized in a broad range of gas flow rates. Beds consisting of large particles tend to fluidize poorly because of bumping and slugging. Even though the fluidization quality can be improved with the addition of fine particles, large particle beds fluidize in a narrow range of gas flow rates. Hence shallower beds must be implemented. In general, gas solid fluidized beds tend to fluidize heterogeneously. However, it is possible to avoid heterogeneity by using low-density particles in a dense gas.

Geldart [50] determined the fluidization properties of various particles through numerous experiments and classified them according to their density and diameter. He determined four groups of particles which are described from smallest to largest particle as follows:

- Group **C**:

  These particles are typically less than 50 microns and are very difficult to fluidize because the interparticle adhesive forces are stronger than those produced by the drag force. These particles will tend to rise as a plug of solids in small-diameter beds and will not fluidize in larger diameter beds. A common approach which is implemented in order to fluidze these particles in larger diameter beds is to inject them into a bed which consists of the same but larger particles. Some examples of group C particles are face powder, flour, and starch.

- Group **A**:

  The size of these particles is typically between 50 and 200 microns and their

density ranges between 700 and 1400 kg/m^3. One of the most important attributes of these particles is that they can fluidize homogeneously at sufficiently low gas flow rates. When the minimum fluidization state is attained in the bed, these particle beds expand considerably from their initial state. This expansion continues at a lower rate until the minimum bubbling velocity, u_{mb}, is reached. In the bubbling state, the bubbles rise more rapidly than the rest of the gas and appear to split and coalesce. The bubbles typically reach a maximum size of approximately 10 cm.

- Group **B**:

  The size of these particles is typically between 40 and 500 microns and their density ranges between 1400 and 4000 kg/m^3. These particles do not undergo homogeneous fluidization. Bubbles begin to form at minimum fluidization. At higher velocities, small bubbles form at the distributor and will grow and coalesce as they propagate through the bed. The size of the bubble is proportional to the relative gas velocity, $u_g - u_{mf}$, and is independent of the particle size.

- Group **D**:

  These are spoutable particles which fluidize poorly in deep beds. If the gas distribution is uneven, these particles will agitate and produce large bubbles or spout. The bubbles coalesce rapidly and become large. They rise more slowly than the rest of the gas. Most applications do not implement such large particles, but in some instances such as in the processing of agricultural products this is unavolidable. Particles belonging to this group require a much greater gas supply than is required for the particular application. This is usually solved by implementing a spouting bed. The 3.3mm alumina catalyst pellets manufactured by the Engelhard Corporation are examples of this type of particle.

Fluidization States

In this section, the homogeneous, bubbling, and slugging fluidization states are discussed in more detail. The turbulent and dilute fluidization states occur more fre-

quently in circulating fluidized beds (CFB) and hence will not be discussed.

As previously stated, the homogeneous state is one in which the particle concentration is nearly uniform. Such a state can only be achieved by group A particles because of their density and diameter. The minimum fluidization velocity is typically predicted by the following equation [51]:

$$\frac{1.75}{\phi_{mf}^3 \Phi_s}\left(\frac{d_p u_{mf}\rho_g}{\mu}\right)^2 + \frac{150(1-\phi_{mf})}{\phi_{mf}^3\Phi_s^2}\left(\frac{d_p u_{mf}\rho_g}{\mu}\right) = \frac{d_p^3\rho_g(\rho_s-\rho_g)g}{\mu^2}, \tag{6.1}$$

where ϕ_{mf} is the porosity at the minimum fluidization state, and Φ_s is the sphericity, which is defined as the ratio of the external surface area of a sphere having the same volume as the particle considered to the external surface area of this particle. The minimum bubbling velocity for group A particles is predicted by [51]

$$u_{mb} = 33 d_p \left(\frac{\rho_g}{\mu}\right). \tag{6.2}$$

The two-phase theory tries to simplify fluidization by breaking it up into two phases. This results in an emulsion state where at a superficial gas velocity u_e and the voidage ϕ_e remain at u_{mf} and ϕ_{mf} respectively. Abrahamson and Geldart found that u_e and ϕ_e change with the inlet gas flow rate u_0. For small solids such as those in group A, they found that these changes could be reasonably represented by

$$\left(\frac{\phi_e}{\phi_{mf}}\right)^3\left(\frac{1-\phi_{mf}}{1-\phi_e}\right) = \left(\frac{u_e}{u_{mf}}\right)^{0.7}. \tag{6.3}$$

In the following section, the expression for the interphase drag force is developed according to [52]. This expression is similar to that in chapter 4 with the velocity of the solid particles accounted for and it is used in the two-phase Navier-Stokes equation model [9] to predict the velocity profiles for the gas and solid phase.

The Drag Force in a Fluidized Bed

An expression for the drag force in a fluidized bed is derived based on a static analysis. The drag force is obtained for a packed bed and is extended to an expression for the drag force of the homogeneous state in a fluidized bed [2].

[2]The discussion in the next two sections is based mainly on ref. [52].

Consider a control volume which consists of a uniform bed of particles of unit cross-sectional area and vertical length L. The power which is dissipated by a fluid flowing through such a control volume is given by

$$\delta E = U \Delta P, \tag{6.4}$$

where U is the volumetric flux and ΔP is the induced pressure drop. The fluid velocity in the control volume is U/ϵ and the energy dissipation rate for a single stationary particle under the influence of a drag force f_d is $U f_d/\phi$. The total number of particles in the bed is $6(1-\phi)L/\pi d_p^3$ and the total rate of energy loss is given by

$$\delta E = \frac{6(1-\phi)L}{\pi d_p^3} \frac{U f_d}{\phi}. \tag{6.5}$$

Substituting equation (6.4) into (6.5) yields

$$f_d = \frac{\pi d_p^3 \phi}{6L(1-\phi)} \Delta P. \tag{6.6}$$

The equation for the pressure drop in a packed bed is given by [52]

$$\Delta P = 60 \frac{\mu_f L U}{d_p^2} \frac{(1-\phi)^2}{\phi^4}, \text{for viscous flow} \tag{6.7}$$

or

$$\Delta P = 1.17 \frac{\rho_f L U^2}{d_p} \frac{(1-\phi)^2}{\phi^4}, \text{for inertial flow.} \tag{6.8}$$

Substituting equation (6.7) into equation (6.6) yields

$$f_d = 3\pi d_p \mu_f U \frac{3.33(1-\phi)}{\phi^3}. \tag{6.9}$$

The term $3.33(1-\phi)/\phi^3$ is called the 'voidage function'. As $\phi \rightarrow 1$, the voidage function becomes zero which is not correct. To remedy this, a 1 is added to the voidage function expression and the correct limit is attained:

$$f_d = 3\pi d_p \mu_f U \left(\frac{3.33(1-\phi)}{\phi^3} + 1 \right). \tag{6.10}$$

Repeating the same analysis for the inertial flow regime yields

$$f_d = 0.055\pi \rho_f d_p^2 U^2 \left(\frac{3.55(1-\phi)}{\phi^3} + 1 \right). \tag{6.11}$$

The voidage functions in equations (6.10) and (6.11) are numerically similar to $\phi^{-3.8}$ and so the equations become

$$f_d = 3\pi d_p \mu_f U \phi^{-3.8}, \tag{6.12}$$

for the viscous flow regime,

$$f_d = 0.055\pi \rho_f d_p^2 U^2 \phi^{-3.8}, \tag{6.13}$$

for the inertial flow regime. These expressions can now be extended to the simple case of a single particle suspension. The *effective weight* of a particle is defined as the net effect of gravity and buoyancy, and is expressed as

$$w_e = \frac{\pi d_p^3}{6}(\rho_p - \rho_f) g \phi. \tag{6.14}$$

The effective weight of the particle can also be interpreted as the drag force which is required to suspend a single, unhindered particle and hence for viscous flow we have

$$w_e = 3\pi d_p \mu_f u_t, \tag{6.15}$$

where u_t is terminal velocity of the particle. Equations (6.12), (6.14), and (6.15) give the drag force for a single particle in the viscous flow regime

$$f_d = \frac{\pi d_p^3}{6}(\rho_p - \rho_f) g \frac{U}{u_t} \phi^{-3.8}. \tag{6.16}$$

The drag force for all flow regimes is given by

$$f_d = \frac{\pi d_p^3}{6}(\rho_p - \rho_f) g \left(\frac{U}{u_t}\right)^{\frac{4.8}{n}} \phi^{-3.8}, \tag{6.17}$$

where n is determined from the Archimedes number, $Ar = g d_p^3 \rho_f (\rho_p - \rho_f)/\mu_f^2$ by

$$\frac{4.8 - n}{n - 2.4} = 0.043 Ar^{0.57}. \tag{6.18}$$

The drag force in a fluidized bed can now be obtained by extension of the drag coefficient relation experienced by a solitary, unhindered particle subjected to a steady fluid velocity U_0

$$f_d = C_D \frac{rho_f U_0^2}{2} \frac{\pi d_p^2}{4}, \tag{6.19}$$

where the drag coefficient C_D is given by

$$C_D = \left(0.63 + \frac{4.8}{Re^{0.5}}\right)^2, Re = \frac{\rho_f U_0 d_p}{\mu_f}. \tag{6.20}$$

Combining equations (6.18), (6.19), and (6.20) yields

$$f_d = \frac{\pi d_p^3}{6}(\rho_p - \rho_f)g\left(\frac{U_0}{u_t}\right)^{\frac{4.8}{n}} \phi^{-3.8}. \tag{6.21}$$

In general, the drag force must be expressed in terms of relative velocities in multiple dimensions, and so equation (6.21) in terms of the drag coefficient becomes

$$f_d = \frac{\pi d_p^3}{6} C_D \frac{3\rho_f(u_f - u_p)|\mathbf{u}_f - \mathbf{u}_p|}{4d_p}\phi^{-1.8}. \tag{6.22}$$

In the following section, we discuss the criterion for the transition from the homogeneous fluidization state to the bubbling state. This criterion provides an allowalble range of inlet velocities which will yield a uniform bed distribution.

Stability Criterion for Fluidized Beds

The stability criterion is used to establish the homogeneous, transition, and the heterogeneous states in a fluidized bed based on the particle density and diameter. When a fluid flux, U, is applied to a bed which is initially fixed at a void fraction, ϵ_1, and the minimum fluidization velocity is only slightly surpassed, the bed first propagates upwards in a piston-like manner. Two interfaces are then established at the top and bottom of the bed. If the top interface is subjected to a small perturbation which displaces a particle into the clear fluid region above, the particle experiences a reduced drag force. This results in a net downward force which returns the particle back to its position. This implies that the top interface is stable, a phenomenon which is also observed experimentally. If the bottom interface is subjected to a small disturbance, the displaced particle also experiences a net downward force. This time the particle is driven further downward and adjacent particles follow as a result of the increase in void fraction in the region abdicated by the original particle. The particles above will fall continuously in this manner causing the particle piston to break down. The bottom interface is thus unstable. The falling particles are stopped at the bottom by

the distributor causing a growing zone of stationary particles at the bottom of the bed. As the particles are falling down from the particle piston, they form a new zone where the void fraction becomes ϵ_2 while the particles above are traveling upwards at ϵ_1. Hence a traveling interface between the ϵ_1 and ϵ_2 zones is created. This interface travels faster than the top surface of the bed. When this interface catches up with the top surface, the rearragement process is complete. The velocity of this interface is given by dL/dt where L is the distance from the bottom of the bed to the interface. By performing a mass balance on the two zones, one obtains a relationship between the interface velocity and the inlet fluxes,

$$\frac{dL}{dt} = (1 - \phi_1)\frac{U_2 - U_1}{\phi_2 - \phi_1}, \tag{6.23}$$

where U_1 is the initial inlet flux, and U_2 is the new inlet flux. Equation (6.23) describes the propagation of a finite discontinuity, or *shock*, separating the two equilibrium states. The formulation of equation (6.23) is based strictly on mass balances and hence assumes that the particles immediately switch from one equilibrium zone to the other one as the shock wave passes over them. No inertial effects are accounted for as the particles would have to slow down before reaching the new equilibrium state. For this condition to hold, the inertial response time for the particles must be negligibly small.

The kinematic wave velocity, u_K, is the limiting value of dL/dt as the amplitude of the imposed fluid flux change $\Delta U = U_2 - U_1$ goes to zero. Hence, we have

$$u_K = (1 - \phi)\frac{dU}{dt}. \tag{6.24}$$

The solids flux has been related to the terminal velocity and void fraction by Lewis et al.[53]

$$U = u_t \phi^n, \tag{6.25}$$

where n is determined from equation (6.18). Substituting equation (6.25) into (6.24) gives

$$u_K = n u_t (1 - \phi)\phi^{n-1}. \tag{6.26}$$

The *dynamic wave* speed, u_D, is a measure of the speed at which inertial effects propagate through the system. When a piston in a cylinder moves forward, it exerts a force on the gas immediately ahead and induces a pressure wave which travels at sonic speed u_D. Under adiabatic conditions,

$$u_D = \sqrt{\frac{\partial p}{\partial \rho_f}}. \tag{6.27}$$

The particle phase in a homogeneously fluidized bed can be thought of as a compressible gas with the distributor being analogous to the piston. An idealized experiment is devised in which the particles are assumed to be arranged in perfectly horizontal layers. The distributor is displaced upwards and the particles in the immediate vicinity are similarly displaced and come in contact with the above adjacent layer. The decrease in local void fraction results in a net force on this adjacent layer causing it to accelerate upwards restoring local equilibrium below it, but exerting a net force on the layer immediately above. This occurs throughout the bed inducing a propagating compression wave analogous to the compressilble fluid case. The pressure impulse is $\delta p \equiv \delta(N_L f)$, where N_L is the number of particles in a layer of unit area and f is the net force experienced by each of the particles as a result of the local void fraction change brought about by the displacement of the distributor. A formula similar to equation (6.34) is applied to determine u_D for the wave traveling through the fluidized bed. Instead of ρ_f, we use $\rho_{pp} = (1-\phi)\rho_p$. This is the 'compressible density' for which the differential form is given by

$$\delta \rho_{pp} \equiv \delta[(1-\phi)\rho_p]. \tag{6.28}$$

Hence, u_D for the solid phase is given by

$$u_D = \sqrt{\frac{\delta p}{\delta \rho_{pp}}} = \sqrt{\frac{\delta(N_L f)}{\delta((1-\phi)\rho_p)}}. \tag{6.29}$$

The concentration wave can be assumed to be the consequence of an imposed perturbation in void fraction, and so equation (6.29) may be expressed by

$$\sqrt{\frac{\partial(N_L f)/\partial\phi}{\partial(1-\phi)\rho_p/\partial\phi}} = \sqrt{-\frac{1}{\rho_p}\frac{\partial(N_L f)}{\partial\phi}}. \tag{6.30}$$

N_L is estimated assuming that the void fraction on the horizontal planes that bisect the particle layers is representative of the average void fraction in the bed

$$N_L = 4(1-\phi)/\pi d_p^2. \tag{6.31}$$

The net primary force acting on a fluidized particle in equilibrium is the sum of the drag and effective weight. Hence, from equations (6.14), (6.22), (6.25), and (6.30) we get

$$\frac{\partial(N_L f)}{\partial \phi} = N_L \frac{\pi d_p^3}{6}(\rho_p - \rho_f) g \left[-3.8 \left(\frac{U}{u_t} \right)^{\frac{4.8}{n}} \phi^{-4.8} - 1 \right] + f \frac{\partial N_L}{\partial \phi}. \tag{6.32}$$

Evaluating the above equation for a void fraction perturbation about the equilibrium condition (f=0, $U = u_t \phi^n$), with N_L given by equation (6.31) and substituting in equation (6.30) gives

$$u_D = \sqrt{3.2 g d_p (1-\phi)(\rho_p - \rho_f)/\rho_p}. \tag{6.33}$$

Using equations (6.26) and (6.33) we arrive at the stability criterion. If $u_D - u_K > 0$, the system is homogeneous; if $u_D - u_K = 0$, the system has reached the stability limit; and if $u_D - u_K < 0$ the system is bubbling.

Heat Transfer Mechanisms in a Fluidized Bed

The main heat transfer mechanisms in a microwave-heated fluidized bed are the heat generated by microwaves within the solid particles, the gas flow around the pellets transporting heat from the bottom to the top of the bed, and solids mixing. It is very difficult to separate the latter two heat transfer mechanisms due to the nature of the fluidization problem, hence they are both accounted for in the convective heat transfer coefficient between the gas and solids. There is considerable controversy among the fluidization researchers as to the correct way to determine the heat transfer coefficient. Some researchers think that the thermal conductivity is not an important contribution to the Nusselt number while others think that accounting for the thermal conductivity is the correct way to determine the Nusselt number and that the Nusselt number

should be above 2. None of the points of view on this matter can be disregarded due to the complexity of the problem.

For Reynolds number which is less than 100, Kothari [54] found that the Nusselt number is a function of only the Reynolds number

$$Nu_{bed} = 0.03Re_p^{1.3}, \tag{6.34}$$

where Re_p is given by

$$Re_p = \frac{d_p u \rho_g}{\mu}. \tag{6.35}$$

The applicability of equation (6.34) has not been proven for particle diameters less than 230μm. However, due to the lack of available data, it is recommended that the above correlation be applied for such particles. It also fails in instances where the solid concentration is too low. In the situation for which the solid concentration is too low and $100 \leq Re_p$, the Ranz correlation is recommended

$$Nu = 2 + (0.6 \rightarrow 1.8)Re_p^{0.5}Pr^{0.33}, \tag{6.36}$$

where $0.6 \rightarrow 1.8$ means that the constant coefficient ranges from 0.6 to 1.8 which are the single sphere and the packed bed limits, respectively. The difference between equations (6.34) and (6.36) are very drastic for Reynolds numbers below 100. Kunii and Levenspiel [49] claim that this is due to formation of clouded bubbles which greatly reduce the contact between the solid and the gas. Gunn [55] derived an expression for gas-solid systems in which the voidage varies between 0.35 and 1:

$$Nu = (7 - 10\phi + 5\phi^2)(1 + 0.7Re_p^{0.2}Pr^{0.33}) + (1.33 - 2.4\phi + 1.2\phi^2)Re_p^{0.7}Pr^{0.33}. \tag{6.37}$$

This correlation predicts results which are similar to the Ranz formula, but has the disadvantage of requiring the knowledge of the overall voidage as function of gas velocity.

Walton et. al [56] found that for coal particles whose size ranges from 0.3 to 0.8mm and for Reynolds numbers ranging from 6 to 50, the Nusselt number is given

by

$$Nu = 0.0028 Re_p^{1.7} \left(\frac{d_p}{D} \right), \tag{6.38}$$

where D is the hydraulic bed diameter. For particle sizes ranging from 0.83 to 8.9mm and for Reynolds numbers ranging from 30 to 120, Siriomiatnikov et. al [57] derived the relation

$$Nu = 0.0097 Re_p Fe^{0.53} \left(\frac{L_{fb}}{d_p} \right)^{-0.45}, \tag{6.39}$$

and for Reynolds numbers ranging from 120 to 2500 the relation

$$Nu = 0.015 Re_p^{0.805} Fe^{0.53} \left(\frac{L_{fb}}{d_p} \right)^{-0.45}, \tag{6.40}$$

where Fe is the Federov number and is given by

$$Fe = 1.1 Ar^{0.33}. \tag{6.41}$$

A more sophisticated model has been developed by Kunii and Levenspiel for fine-particle bubbling beds from mass transfer considerations. The heat transfer coefficient, h_{bc}, at the bubble-cloud interface is given by

$$h_{bc} = 0.975 \rho_g C_{pg} \left(\frac{k_g}{\rho_g C_{pg}} \right)^{0.5} \left(\frac{g}{d_b} \right)^{0.25}. \tag{6.42}$$

Based on a unit volume of bubble phase, the overall heat transfer coefficient across the bubble-cloud boundary is

$$H_{bc} = \frac{v C_{pg} + h_{bc} S_{bc}}{V_b} = 4.5 \left(\frac{u_{mf} \rho_g C_{pg}}{d_b} \right) + 5.85 \frac{(k_g \rho_g C_{pg})^{0.5} g^{0.25}}{d_b^{1.25}}, \tag{6.43}$$

where v is the volumetric flow of gas from bubble to cloud, and S_{bc} is the surface area of the bubble cloud. Since H_{bc} does not account for the heat transfer from the particles within the bubble to the bed of solids, the overall convective heat transfer coefficient is given by

$$H_{tot} = \gamma_b \frac{6 Nu k_g}{\Phi_s d_p^2} \eta_h + H_{bc}, \tag{6.44}$$

where γ_b is the volume of solid dispersed within the bubble. From the definition of the heat transfer coefficient

$$\delta H_{tot} = \frac{6}{\Phi_s d_p}(1-\epsilon)h_{bed} \tag{6.45}$$

we get the bed Nusselt number by substituting equation (6.43) and (6.44) into equation (6.45)

$$Nu_{bed} = \frac{h_{bed}d_p}{k_g} = \frac{\delta}{1-\phi}\left[\gamma_b Nu\eta_h + \frac{\Phi_s d_p^2}{6k_g}H_{bc}\right], \tag{6.46}$$

where δ is the bubble fraction in a fluidized bed and is given by

$$\delta = \left(\frac{1-\phi}{1-\phi_{mf}}\right)\left(\frac{u_0 - u_m f}{u_{br}}\right), \tag{6.47}$$

where u_{br} is the velocity at which the bubble rises. The heat transfer adsorption efficiency, η_h is given by

$$\eta_h = \frac{1}{1+\alpha\frac{\rho_g C_{pg}}{\rho_s C_{ps}}}. \tag{6.48}$$

For fine particles, α ≅20-1000, and so η_h ≅0.98-0.91. The problem with the above formulation is that it relies on many intermediate relationships that may yield incorrect results. Knowledge of the spacing between grid holes, the bubble size and rise velocity are required to implement this correlation.

Thermo-Fluid Model of the Fluidized Bed

Fluidized beds are categorized as multiphase flow problems. As mentioned in chapter two, there are currently five approaches to model multiphase flow problems. The best overall balance between computational time and accuracy seems to be achieved by implementing the volume-averaging or Eulerian-Granular approach. Assumptions 1, 6, and 8 of the packed bed model hold for the fluidized bed as well. In addition we assume

1) temperature dependence of the fluid properties is neglected,

2) the bed is 2-d and rectangular,

3) the velocity and volume fraction profile is assumed to be the time-averaged values over a specified time interval,

4) only homogeneous fluidization is considered,

5) only the expanded bed region is considered in the analysis, i.e., the outflowing gas is not analyzed,

6) compressibility and viscous heating effects are neglected, and

7) the electric field can vary linearly in the flow direction.

The fluid mechanics portion of the analysis was performed with FLUENT 6.1 [9]. FLUENT is a commercial CFD software package which provides capabilities for solving multiphase fluid flow problems. The multiphase fluid flow problem is very complicated and involved and would be difficult to solve otherwise. FLUENT simplifies this by merely requiring the user to provide the solid and fluid properties as well as the boundary conditions. The boundary conditions for the fluidized bed considered in this work are as follows: 1) the gas and solid velocities as well as the void fraction are specified at the bed inlet. The solid velocities are set to zero and the void fraction is 1. The inlet gas velocity is also specified; 2) The no-slip condition was implemented at the walls for both phases; 3) The pressure is specified at the bed outlet. For all cases, atmospheric pressure was assumed.

The detailed equations which are solved by FLUENT will not be discussed here. For details, see reference [58]. To solve the multiphase Navier-Stokes equations, the viscosity and pressure for the solid phase must be quantified. This is accomplished in FLUENT by implementing a kinetic theory approach in which the solid particles are treated as gas molecules and the granular temperature of the solid phase is computed. The granular temperature is a measure of the solid phase chaos within the bed and the solids pressure and viscosity are functions of this quantity. The granular temperature equation is a differential equation which is derived from the Boltzman equation. FLUENT requires the user to specify the granular temperature at the inlet. FLUENT's default value for the granular temperature at the inlet is 10^{-4} and is retained for all cases.

The system of equations and constitutive relations required to ultimately solve

the multiphase fluid flow problem is computationally intensive. It could easily take several weeks to fully analyze one case. The reason that this occurs is because the fluidized bed requires a transient analysis with time steps no greater than 10^{-3}. In some cases [7], time steps as small as 10^{-4} or even 10^{-5} are necessary. For this reason, an approximation must be made in order to decrease the computational time. In several discussions with the fluidization group in the chemical engineering department at The Illinois Institute of Technology (IIT), it is known that fluidized beds reach an oscillating "steady state" after approximately 10-15 seconds. The group at IIT has done extensive research in fluidization technology since the early 60's. Today, most of the research in this area is carried out by students under the guidance of professors Dimitri Gidaspow and Hamid Arastoopour. Professors Gidaspow and Arastoopour have authored many articles in the area of fluidization. To get a list of many of their publications, see reference [7].

Hence, all relevent quantities such as solid and gas velocities and void fractions are time-averaged over a specific time interval. The time intervals for specific cases are discussed in the next chapter. The resulting time-averaged quantities are then implemented in a thermal model. The required computational effort is greatly reduced because the heat transfer analysis must be performed on minute time scales whereas the fluid mechanics codes are designed for second time scales.

A FORTRAN code was developed for the purpose of computing the temperature distribution in the bed. The time-averaged values from the fluid mechanics analysis were used as inputs for the heat transfer code. The equations governing the temperature distribution in the bed are given by

$$\phi_s \rho_s C_{ps} \frac{\partial T_s}{\partial t} + C_{ps} \frac{\partial}{\partial x}(\phi_s \rho_s u_s T_s) + C_{ps} \frac{\partial}{\partial y}(\phi_s \rho_s v_s T_s) \tag{6.49}$$
$$= \frac{\partial}{\partial x}\left(\phi_s k_s \frac{\partial T_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(\phi_s k_s \frac{\partial T_s}{\partial y}\right) + h_v(T_f - T_s) + \phi_s \dot{q},$$

and

$$\phi_f \rho_f C_{pf} \frac{\partial T_f}{\partial t} + C_{pf} \frac{\partial}{\partial x}(\phi_f \rho_f u_f T_f) + C_{pf} \frac{\partial}{\partial y}(\phi_f \rho_f v_f T_f) \tag{6.50}$$
$$= \frac{\partial}{\partial x}\left(\phi_f k_f \frac{\partial T_f}{\partial x}\right) + \frac{\partial}{\partial y}\left(\phi_f k_f \frac{\partial T_f}{\partial y}\right) + h_v(T_s - T_f),$$

where T_s is the solid temperature, T_f is the gas temperature, and h_v is the volumetric heat transfer coefficient which is obtained from equation (6.34). The boundary conditions for the solid phase are

$$-\phi_s k_s \frac{\partial T_s}{\partial x}\Big|_0 = U(T_s - T_\infty)|_0, \quad -\phi_s k_s \frac{\partial T_s}{\partial x}\Big|_L = U(T_s - T_\infty)|_L, \tag{6.51}$$

and

$$\frac{\partial T_s}{\partial y}\Big|_0 = \frac{\partial T_s}{\partial y}\Big|_H = 0, \tag{6.52}$$

where L is the length of the bed, H is the bed height, U is given by equation (4.30), and R_t is given by,

$$R_{tot} = \frac{L}{k_g A} + \frac{1}{hA}, \tag{6.53}$$

where h is defined by equation (4.32). For the gas phase,

$$-\phi_f k_f \frac{\partial T_f}{\partial x}\Big|_0 = U(T_f - T_\infty)|_0, \quad -\phi_f k_f \frac{\partial T_f}{\partial x}\Big|_L = U(T_f - T_\infty)|_L, \tag{6.54}$$

$$T_f(0) = T_{in}, \tag{6.55}$$

and

$$\frac{\partial T_f}{\partial y}\Big|_H = 0. \tag{6.56}$$

The solution procedure for the energy equations is similar to that described in chapter 4. The approach is described in chapter 5 of [48]. A brief outline is given for the solid phase. The gas phase is similar. Equation (6.49) can be rewritten in the form

$$\rho_s \phi_s \frac{\partial T_s}{\partial t} + \frac{\partial J_{sx}}{\partial x} + \frac{\partial J_{sy}}{\partial y} - \frac{h_v}{Cp_s}(T_f - T_s) = S_s, \tag{6.57}$$

where,

$$J_{sx} = \phi_s \rho_s u_s T_s - \Gamma_s \frac{\partial T_s}{\partial x}, \tag{6.58}$$

$$J_{sy} = \phi_s \rho_s v_s T_s - \Gamma_s \frac{\partial T_s}{\partial y}, \tag{6.59}$$

$$\Gamma_s = \frac{\phi_s k_s}{Cp_s}, \tag{6.60}$$

$$S_s = \frac{\phi_s}{Cp_s} \dot{q}_s. \tag{6.61}$$

Let $\vec{F} = \phi_s \rho_s \vec{V}_s$, $D_s = rho_s/\delta x$, and $\vec{P} = \vec{F}/D_s$, where $\vec{V}$ is the velocity vector. The discretized forms of these, with e, w, n, and s notation as in chapter 4, become

$$F_{se} = (\rho_s \phi_s u_s)_e \Delta y,\ D_{se} = \frac{\Gamma_e \Delta y}{\Delta x},\ P_{se} = \frac{F_{se}}{D_{se}}, \tag{6.62}$$

$$F_{sw} = (\rho_s \phi_s u_s)_e \Delta y,\ D_{sw} = \frac{\Gamma_w \Delta y}{\Delta x},\ P_{sw} = \frac{F_{sw}}{D_{sw}}, \tag{6.63}$$

$$F_{sn} = (\rho_s \phi_s v_s)_e \Delta x,\ D_{sn} = \frac{\Gamma_n \Delta x}{\Delta y},\ P_{sn} = \frac{F_{sn}}{D_{sn}}, \tag{6.64}$$

$$F_{ss} = (\rho_s \phi_s v_s)_e \Delta x,\ D_{ss} = \frac{\Gamma_s \Delta x}{\Delta y},\ P_{ss} = \frac{F_{ss}}{D_{ss}}. \tag{6.65}$$

The discretized form of equation (6.49) is

$$a_{Ps} T^1_{Ps} - a_{Es} T^1_{Es} - a_{Ws} T^1_{Ws} - a_{Ns} T^1_{Ns} - a_{Ss} T^1_{Ss} - a_{Pf} T^1_{Pf} = b, \tag{6.66}$$

where

$$a_{Es} = D_{se} A(|P_{se}|) + max[-F_{se}, 0], \tag{6.67}$$

$$a_{Ws} = D_{sw} A(|P_{sw}|) + max[-F_{sw}, 0], \tag{6.68}$$

$$a_{Ns} = D_{sn} A(|P_{sn}|) + max[-F_{sn}, 0], \tag{6.69}$$

$$a_{Ss} = D_{ss} A(|P_{ss}|) + max[-F_{ss}, 0], \tag{6.70}$$

$$a_{Pf} = \frac{h_{Pv}}{Cp_{Ps}} \Delta x \Delta y, \tag{6.71}$$

$$a^0_{Ps} = \frac{\rho_s \phi_{Ps}}{\Delta t} \Delta x \Delta y, \tag{6.72}$$

$$a_{Ps} = a_{Es} + a_{Ws} + a_{Ns} + a_{Ss} + a_{Pf} + a^0_{Ps}, and \tag{6.73}$$

$$b = \frac{\phi_{Ps}}{Cp_{Ps}} \dot{q}^0_{Ps} \Delta x \Delta y + a^0_{Ps} T^0_P, \tag{6.74}$$

where $A(|P_{se}|)$, $A(|P_{sw}|)$, $A(|P_{sn}|)$, and $A(|P_{ss}|)$ are determined by the numerical scheme to be implemented. The possible choices in [48] are the central difference, upwind, hybrid, and power law schemes. The upwind scheme was chosen in all analyses. For more details on these schemes, see [48].

Chapter 7

Results for the Fluidized Bed

In this study, air flow through a fluidized bed consisting of gamma alumina particles heated by microwaves is examined. The effects of the particle diameter and inlet velocity on the bed temperature and the temperature difference between the phases are analyzed. These effects are observed for particle diameters of 100, 125, and 150 μm. The nominal minimum fluidization velocities of these particles as predicted by equation (6.1) are 33, 51, and 73 mm/s respectively. The simulations were performed at 6 times the nominal minimum fluidization velocity for each particle diameter and at 4 times the nominal minimum fluidization velocity for the 150 micron particles. The hypothetical container has a length of 40 cm and a height of 60 cm as shown in figure 2 of chapter 1. Initially, the container is filled to 1/4 of its height with solid particles. For the hydrodynamic portion of the computation, the bed was subdivided into a 50 by 50 grid with a time step of 5×10^{-4} seconds. The grid size and time step were determined by the parameter $F = \Delta t/(\Delta x)^2$ [59]. The value for F was chosen comparable with those in reference [7]. Typical values for F in reference [7] were about 2.5, and the value of F for the current grid is 7.8. Also, the y-direction is chosen as the flow direction. As mentioned in the previous chapter, the hydrodynamic analysis is decoupled from the thermal analysis. This allows the flexibility of choosing a different time step for the thermal analysis. Also, the x and y components of velocity for both phases as well as the solid volume fraction were time-averaged over an interval of 90-100 seconds for the 6 times minimum fluidization cases and 120-130 seconds for

the 4 times minimum fluidization case. The expanded bed height ranged from 21.6 to 26.4 cm after the application of the time-averaging for all cases. This resulted in a grid which ranged from 50 cells in the x-direction by 18 to 21 cells in the y-direction for the thermal analysis. A time step of 10 seconds was chosen for the thermal analysis after a comparison of the results between a 10 and 0.1 seconds time step yielded a 1.2 percent error for the 150 micron particles at 6 times nominal minimum fluidization. The air enters the bed at 300.15 K. In the following figures, the standard 2-dimensional, right-hand cooridinate system is assumed, i.e., +x points towards the right of the page and +y points up and is the flow direction. The color scale in the contour plots does not vary with time, but the magnitude associated with this scale varies with time.

Figures 7.1-7.3 display the time evolution of the solid x and y-components of velocity for the 100 micron particles at 6 times the nominal minimum fluidization velocity at 10 second intervals. It can be seen that the flow achieves a steady pattern after 40 seconds.

Figures 7.3-7.6 display similar behaviour for 150 micron particles at 6 times nominal minimum fluidization. For this case, it can be seen that a steady pattern is achieved after 50 seconds. The solid particles appear to fall in the center and next to the walls of the bed in both cases. The solids appear to rise in the region right next to the center and next to the falling region near the walls. The 150 micron particles rise closer to the wall than do the 100 micron particles. It can be seen from the figures that the velocity profiles are slightly asymmetrical even though the boundary conditions are identical at the walls and the initial condition is a uniform packed bed. It is believed by FLUENT's [9] multiphase flow staff that this could be caused by perturbations in the void fraction which occur after the third or fourth significant figure. These perturbations induce asymmetries in the void fraction which in turn induce asymmetries in the drag force because the drag force is inversely proportional to the cube of the void fraction. Such anomalies have also been encountered by previous authors [52, 60-62]. While it is believed [9] that the asymmetries represent actual physical phenomena, the reasons for such asymmetries

have not been determined and are still under investigation.

Figures 7.7-7.10 display the time-averaged distributions of the x and y-components of velocity for the gas and solid phase as well as the solid volume fraction for the four cases mentioned above. These figures reveal that the y-component of velocity is dominant throughout and that the solid volume fraction is constant throughout most of the bed. An interesting observation is that for the 100 micron particles the solids flow in the same direction relative to the gas while the solids flow in the opposite direction relative to the gas towards the center of the bed in the remaining cases. Also, it can be seen that the y-component of velocity is nearly symmetrical while the x-component is assymetrical in the current cases.

Figures 7.11-7.14 display the time evolution of the x and y-components of solid velocity as well as the solid volume fraction over the time-averaging interval for a point located in the middle of the bed for each case. The time evolution was plotted for nine points in the bed for each case, but due to the similarity of the plots it is sufficient to display the plots for the point located at the center of the bed. It can be seen that that the quantities in question show very small fluctuations and hence time-averaging is a reasonable approximation.

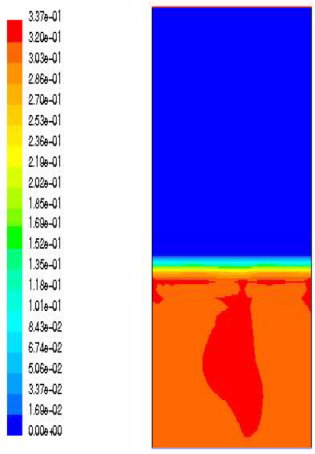
Figures 7.15-7.22 display the time evolution of the temperature, the steady-state temperature profiles, and the temperature difference between the gas and solid phases for the cases mentioned above. Just as for the packed bed, the points for the horizontal profiles are located at half of the bed height and the points for the vertical profiles are located halfway between the center and the wall of the bed. As expected, the temperature increases with time along the horizontal and vertical directions. The steady-state profiles reveal that the temperature increases towards the center of the bed near the inlet and towards the right away from the inlet for the 100 micron particles and decreases towards the center near the inlet for the remaining cases at 6 times the nominal minimum fluidization velocity. The variation in temperature in the x-direction is more pronounced at the inlet than away from the inlet. This behaviour near the inlet could be due to a combination of the fact that the solid is heated at the inlet while the gas maintains a constant temperature there and the

y-component of velocity of the solid phase having the same direction as the gas phase throughout the bed. This does not allow the gas to remove as much energy from the solid as would be possible if the solids were either stationary or moving in the opposite direction because of the lower relative velocity between the gas and solid phases. The nonmonotonicity in the vertical direction, and the off-center maximum temperature in the x-direction can be attributed to the distribution of the x and y-velocity components of the gas and solid phases. At certain points in the bed, the x-component of velocity may be as much as 30 times smaller than its neighbors and hence less energy enters than exits a control volume surrounding such points. The y-component of velocity is nonmonotonic in the horizontal and vertical directions and hence more heat is removed in some regions than in others. The asymmetrical distribution of the x-component of velocity, shown in figure 7.9, could be a large contributing factor in the asymmetrical temperature profile in the 150 micron case at 4 times the nominal minimum fluidization velocity.

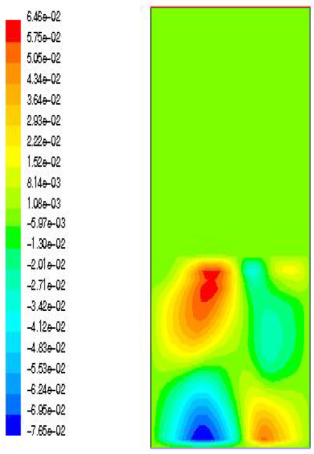
All computations discussed so far have been performed with a uniform electric field of 3 kV/m. In order to induce a greater temperature difference between the solid and gas throughout the bed, the rate of heat generation must be increased as the distance away from the inlet increases. One way to do this is to assume a nonuniform electric field in the y-direction. Hence, a computation was performed for the 150 micron particle case at 4 times the nominal minimum fluidization velocity where the electric field is assumed to vary linearly in the flow direction. The electric field is described by the equation $E = ay + E_0$, where E is the electric field, a is a constant whose value is taken to be $10^5 V/m^2$, y is the location at which the electric field is determined, and E_0 is the electric field at the inlet, taken to be 2 kV/m. Figure 23 shows the steady-state temperature, and the temperature difference between the gas and solid phase. Qualitatively, the steady-state plots are similar to those in figure 20 with higher temperatures. Once again, the maximum temperature is off-center for both phases. The temperature difference between the gas and solid reach 300 K on the right boundary and is asymmetrical at the inlet.

In general, problems which involve flow reversal have many things occuring

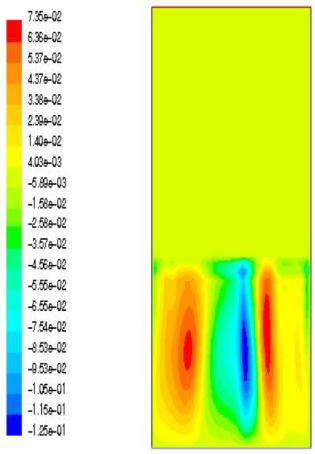
simultaneously. The flow downstream is influenced by the flow upstream and vice versa. This makes it very difficult to explain the physics in these types of problems.



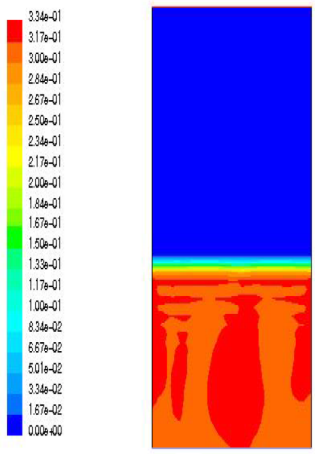
Solid volume fraction at 30 seconds.



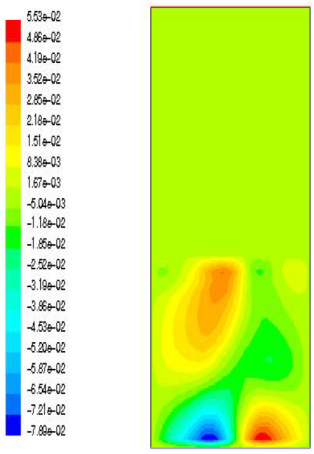
Solid x-velocity at 30 seconds.



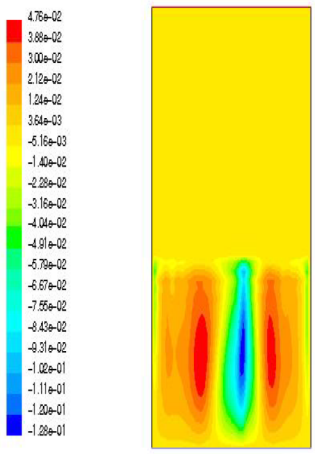
Solid y-velocity at 30 seconds.



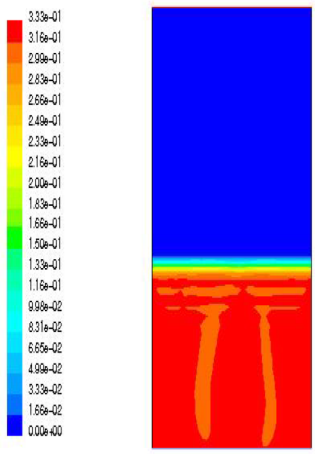
Solid volume fraction at 40 seconds.



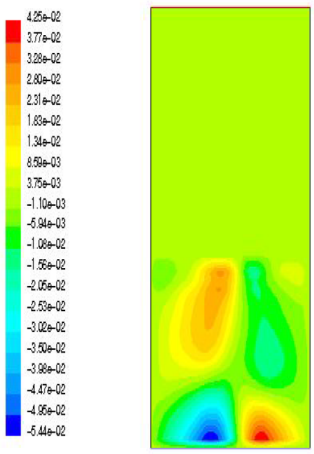
Solid x-velocity at 40 seconds.



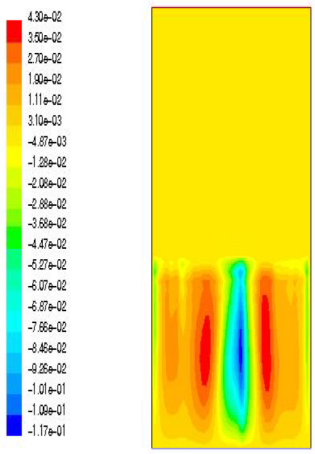
Solid y-velocity at 40 seconds.



Solid volume fraction at 50 seconds.

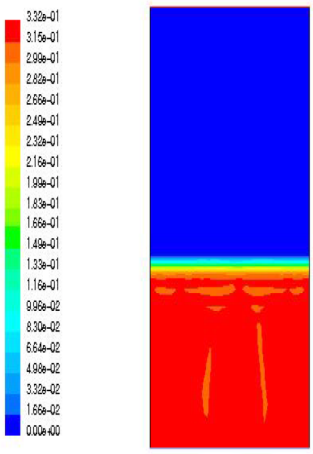


Solid x-velocity at 50 seconds.

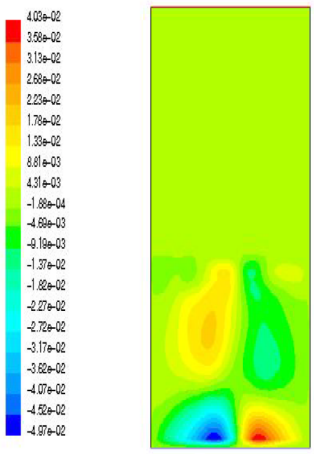


Solid y-velocity at 50 seconds.

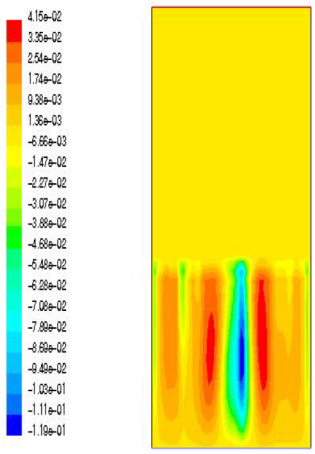
Figure 7.1: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 100 micron particles at 6 the times nominal minimum fluidization. Note that the scale is relative.



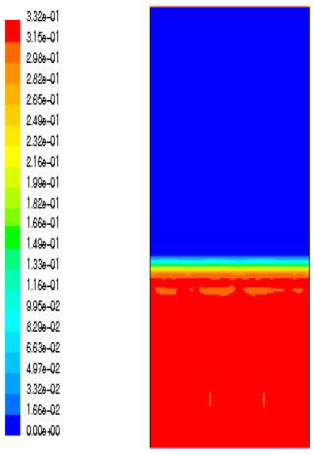
Solid volume fraction at 60 seconds.



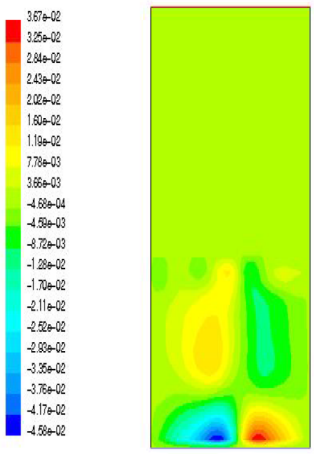
Solid x-velocity at 60 seconds.



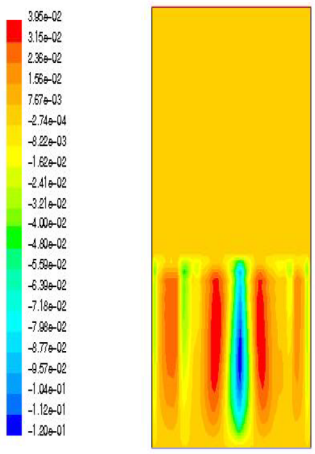
Solid y-velocity at 60 seconds.



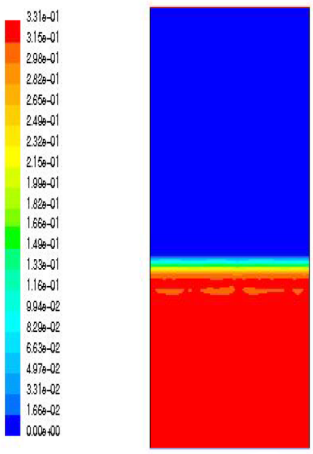
Solid volume fraction at 70 seconds.



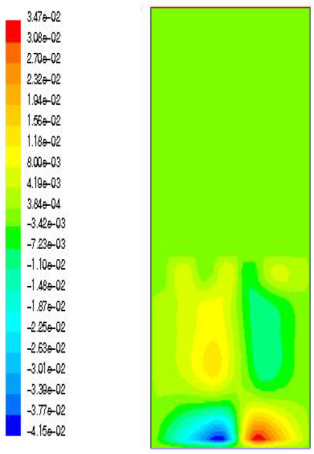
Solid x-velocity at 70 seconds.



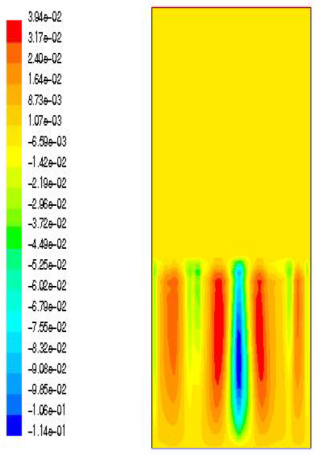
Solid y-velocity at 70 seconds.



Solid volume fraction at 80 seconds.

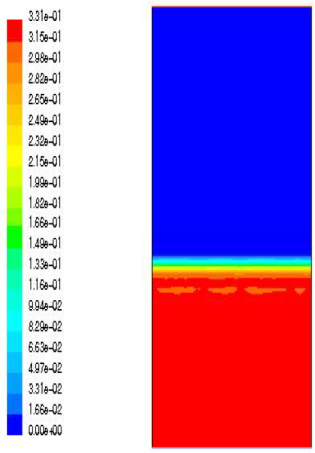


Solid x-velocity at 80 seconds.

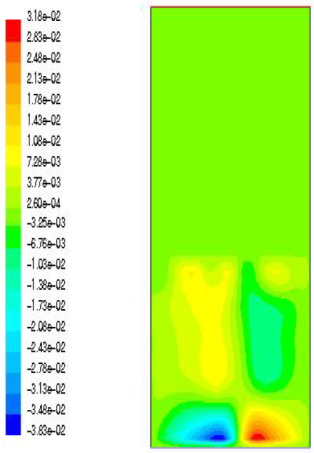


Solid y-velocity at 80 seconds.

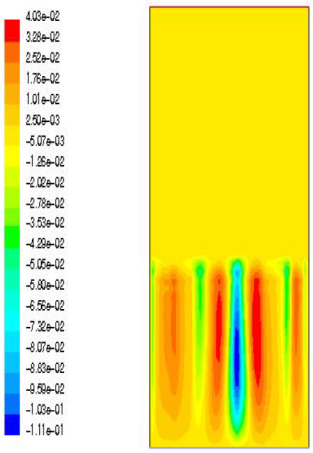
Figure 7.2: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 100 micron particles at 6 times the nominal minimum fluidization. Note that the scale is relative.



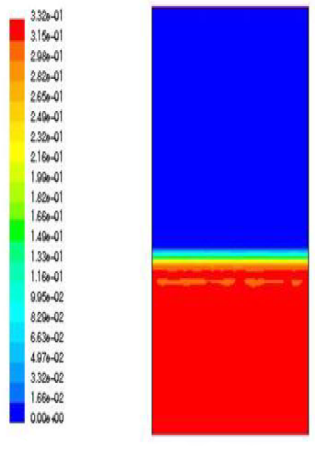
Solid volume fraction at 90 seconds.



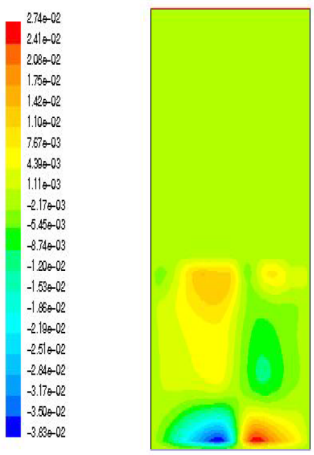
Solid x-velocity at 90 seconds.



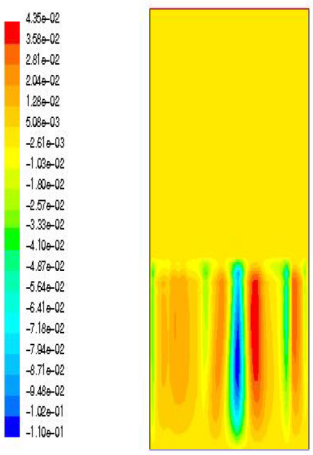
Solid y-velocity at 90 seconds.



Solid volume fraction at 100 seconds.

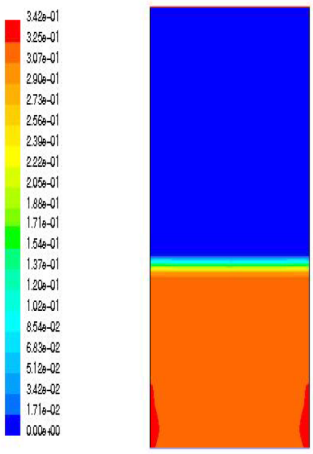


Solid x-velocity at 100 seconds.

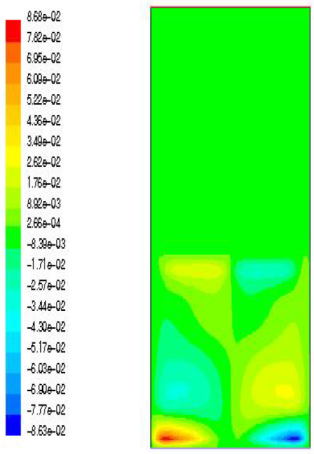


Solid y-velocity at 100 seconds.

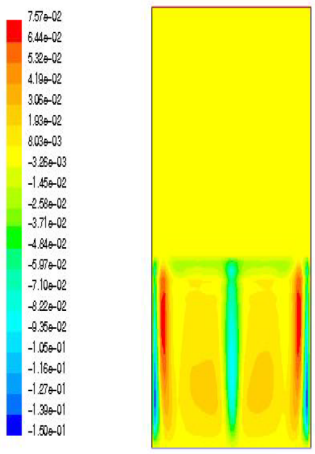
Figure 7.3: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 100 micron particles at 6 times nominal minimum fluidization. Note that the scale is relative.



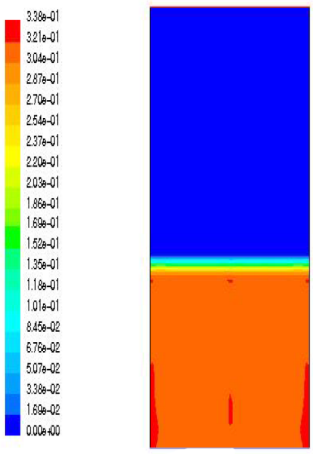
Solid volume fraction at 30 seconds.



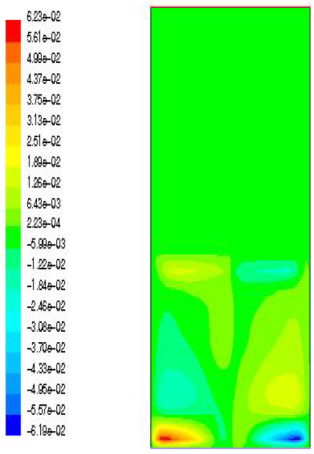
Solid x-velocity at 30 seconds.



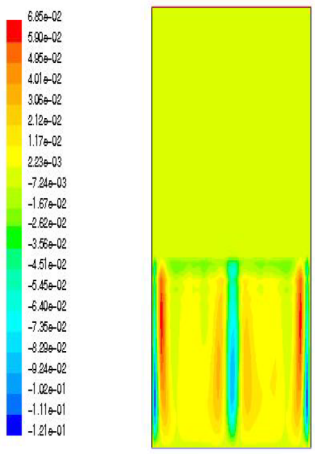
Solid y-velocity at 30 seconds.



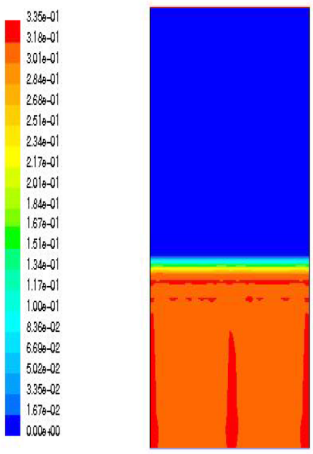
Solid volume fraction at 40 seconds.



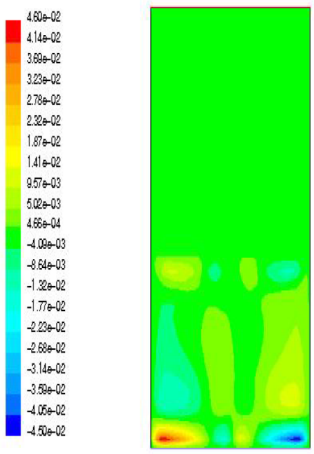
Solid x-velocity at 40 seconds.



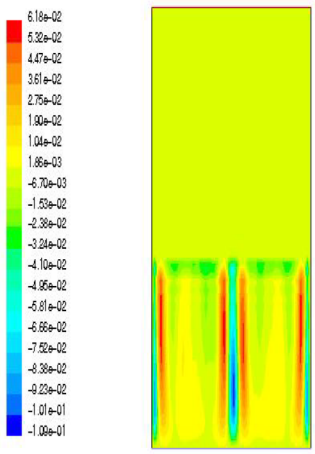
Solid y-velocity at 40 seconds.



Solid volume fraction at 50 seconds.

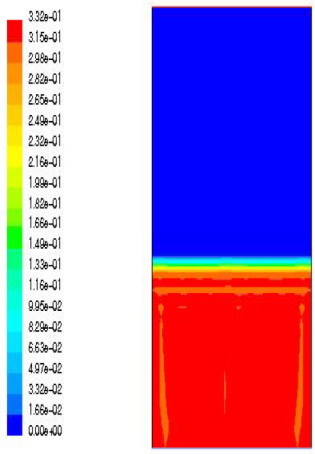


Solid x-velocity at 50 seconds.

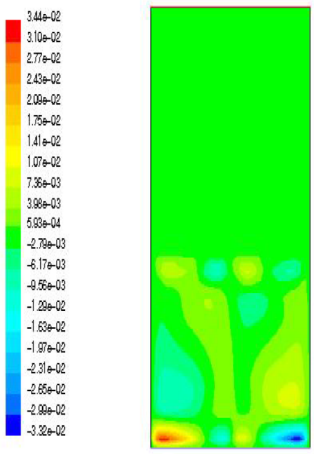


Solid y-velocity at 50 seconds.

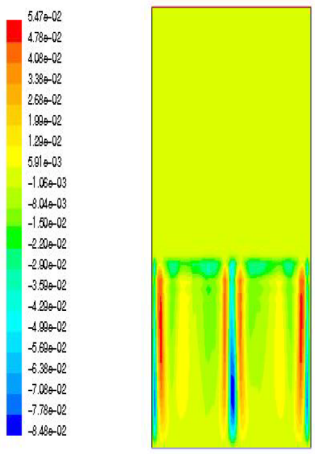
Figure 7.4: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 150 micron particles at 6 times nominal minimum fluidization. Note that the scale is relative.



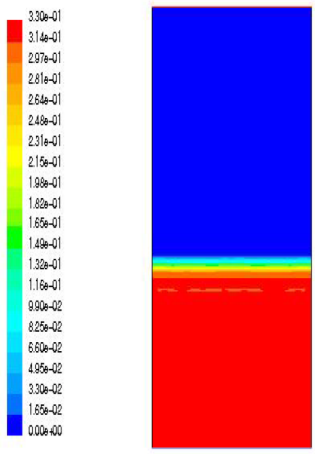
Solid volume fraction at 60 seconds.



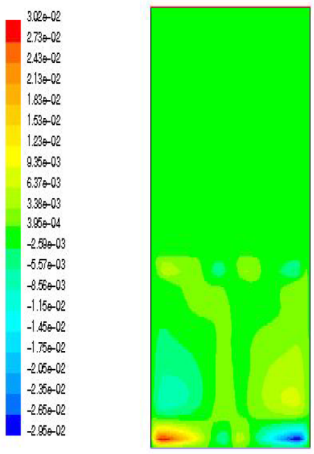
Solid x-velocity at 60 seconds.



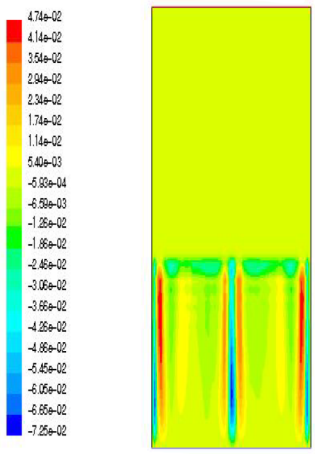
Solid y-velocity at 60 seconds.



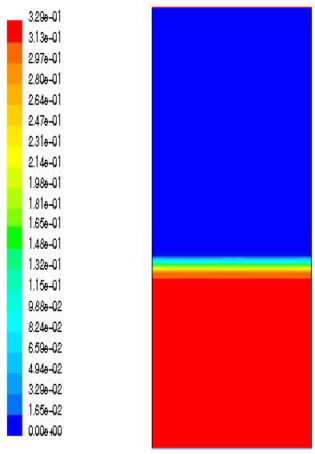
Solid volume fraction at 70 seconds.



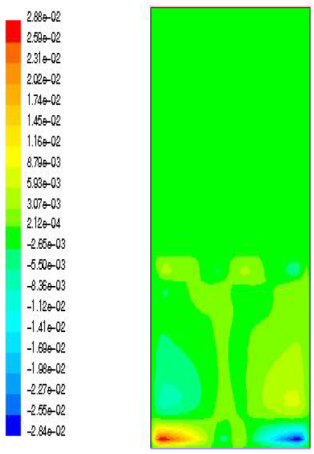
Solid x-velocity at 70 seconds.



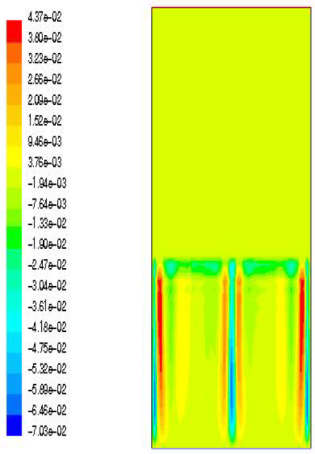
Solid y-velocity at 70 seconds.



Solid volume fraction at 80 seconds.

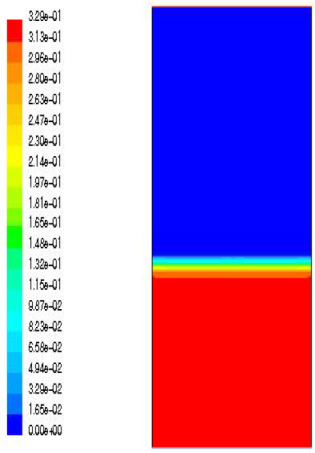


Solid x-velocity at 80 seconds.

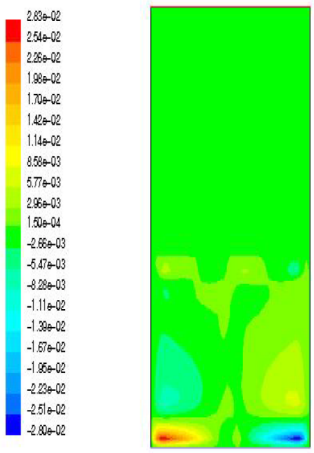


Solid y-velocity at 80 seconds.

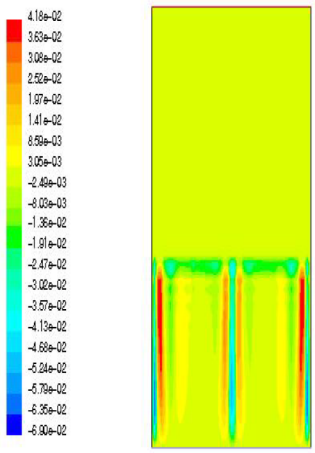
Figure 7.5: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 150 micron particles at 6 times nominal minimum fluidization. Note that the scale is relative.



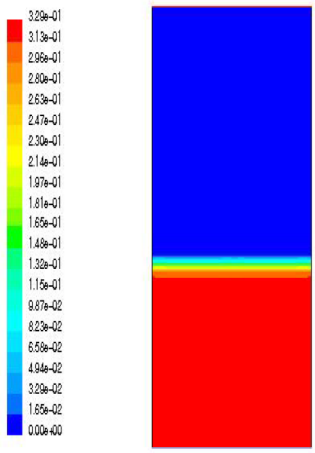
Solid volume fraction at 90 seconds.



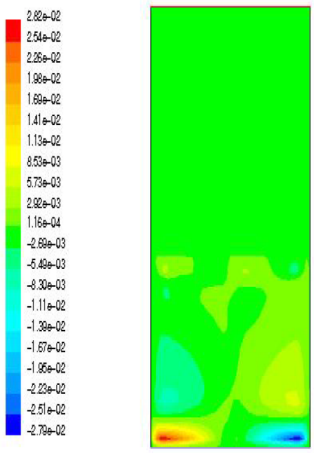
Solid x-velocity at 90 seconds.



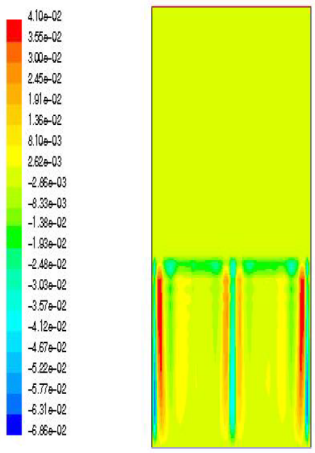
Solid y-velocity at 90 seconds.



Solid volume fraction at 100 seconds.



Solid x-velocity at 100 seconds.



Solid y-velocity at 100 seconds.

Figure 7.6: Time evolution of the solid volume fraction, and the x, and y-components of solid velocity for the 150 micron particles at 6 times nominal minimum fluidization. Note that the scale is relative.

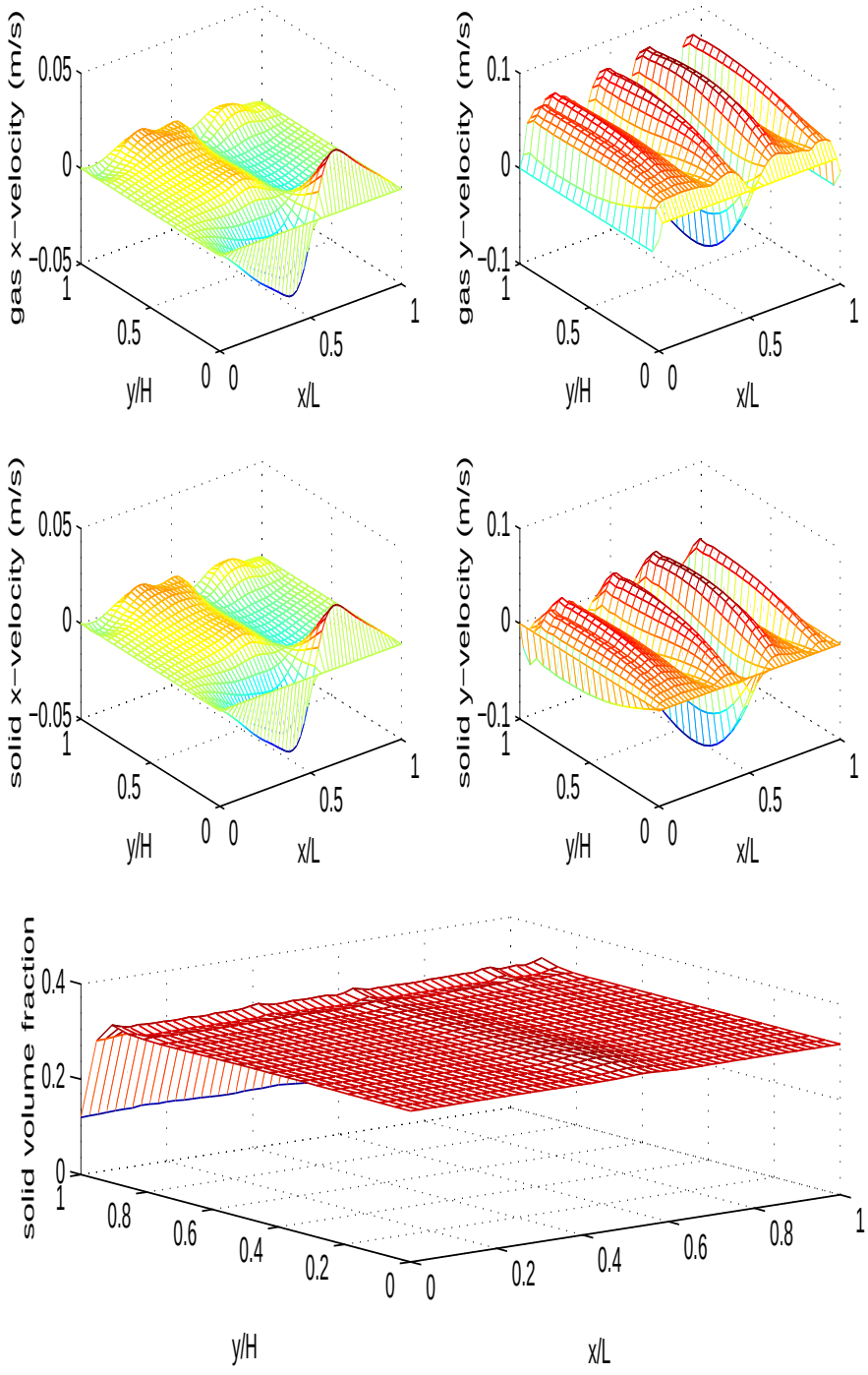


Figure 7.7: Time-averaged x and y-components of velocity as well as the solid volume fraction for 100 micron particles at 6 time minimum fluidization.

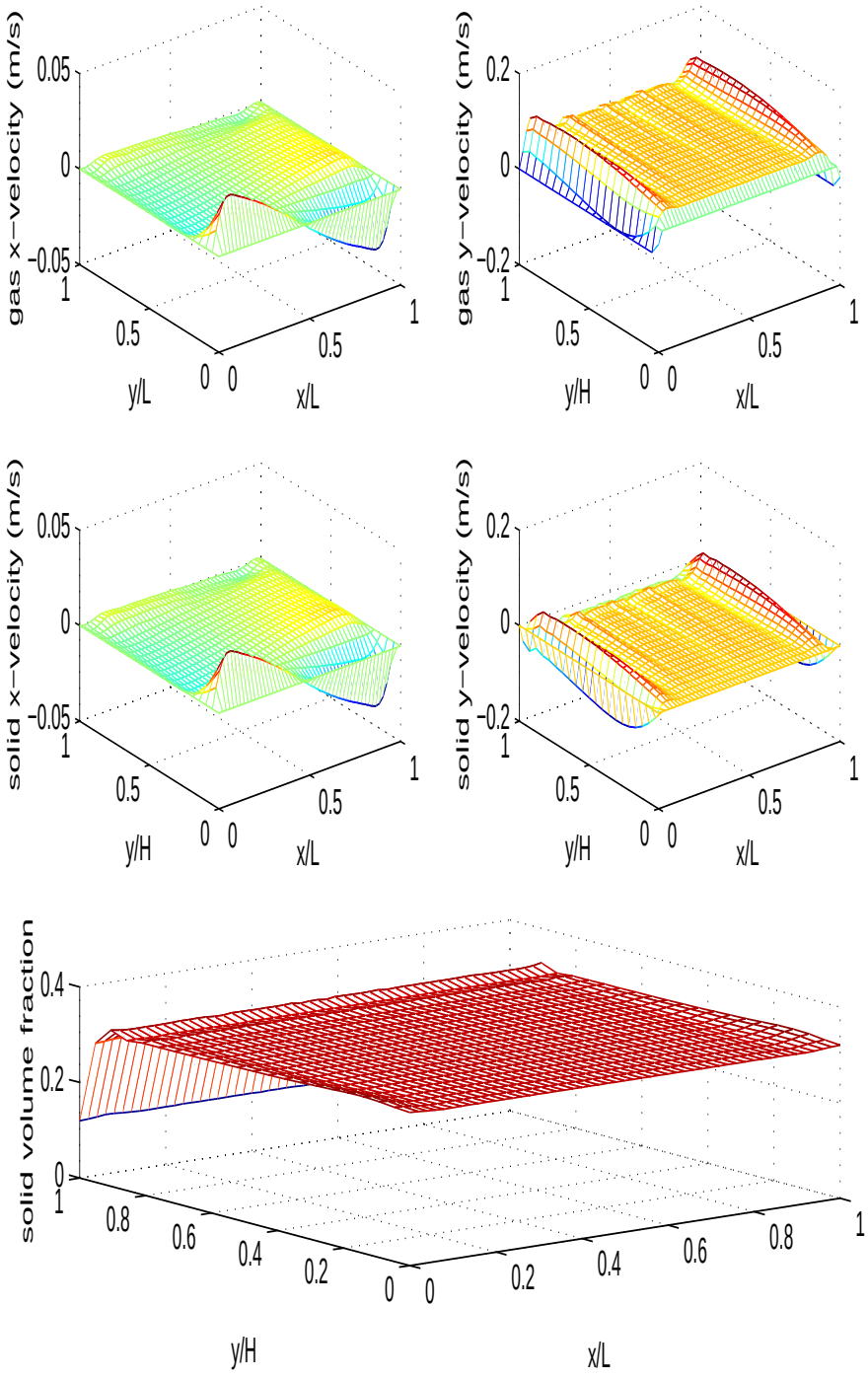


Figure 7.8: Time-averaged x and y-components of velocity as well as the solid volume fraction for 125 micron particles at 6 time minimum fluidization.

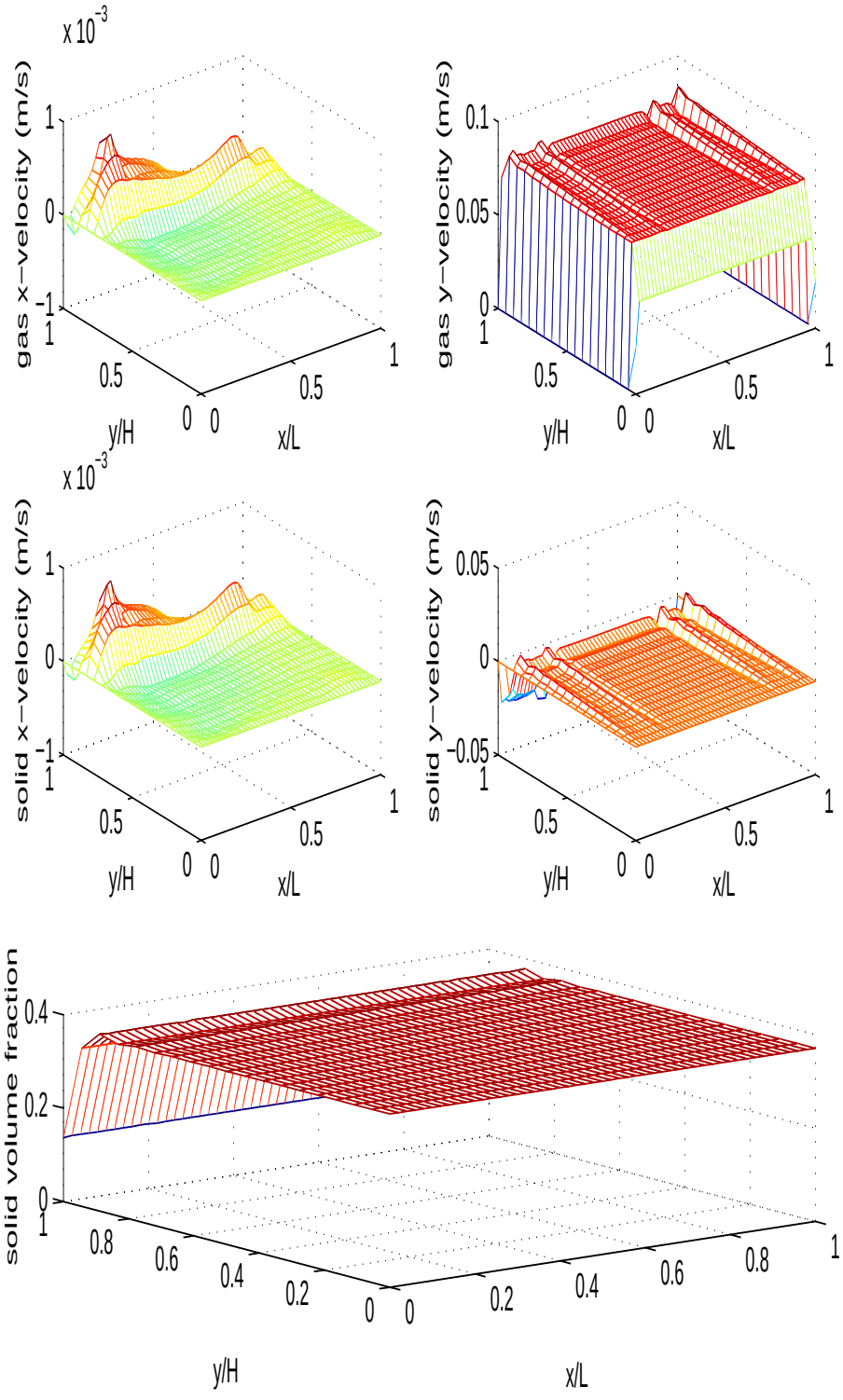


Figure 7.9: Time-averaged x and y-components of velocity as well as the solid volume fraction for 150 micron particles at 4 time minimum fluidization.

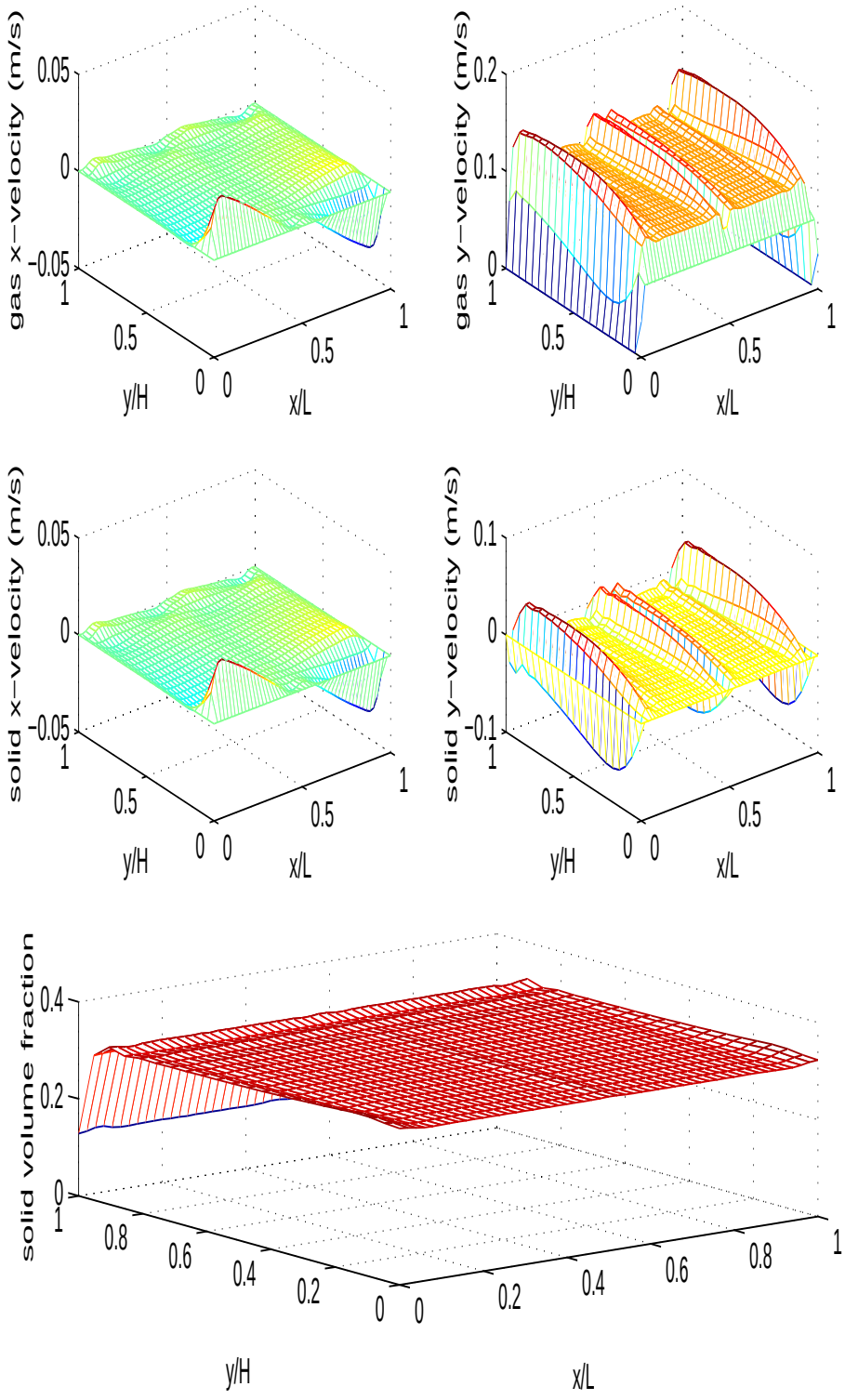


Figure 7.10: Time-averaged x and y-components of velocity as well as the solid volume fraction for 150 micron particles at 6 time minimum fluidization.

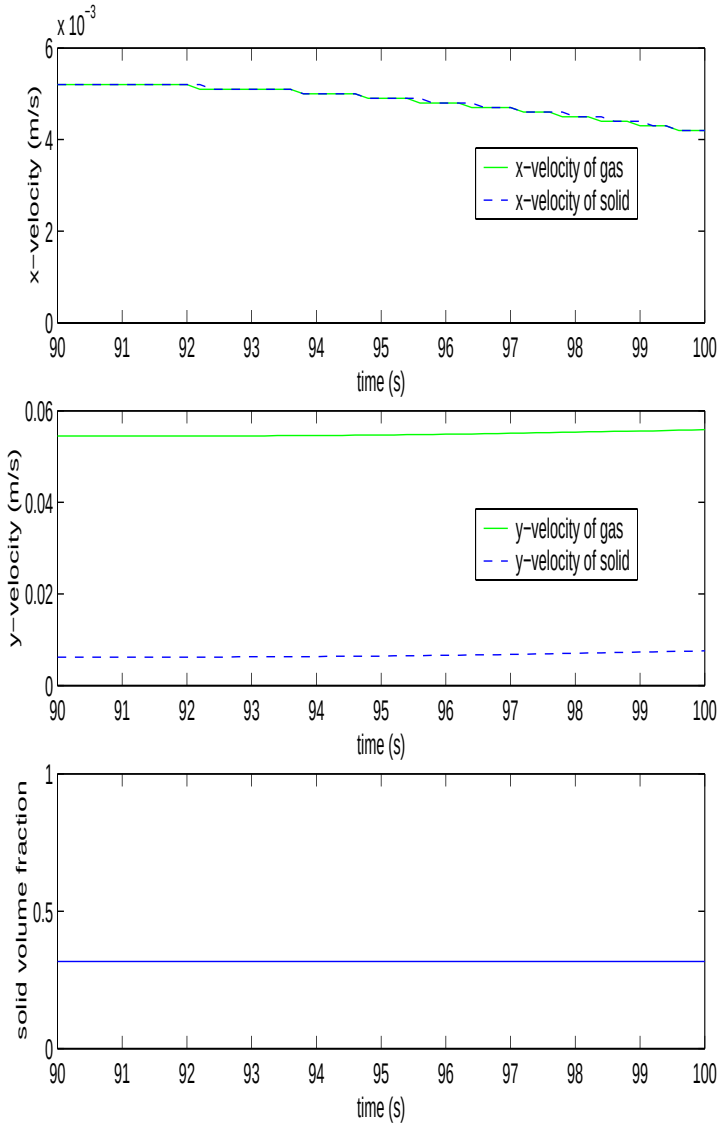


Figure 7.11: Time evolution of x and y-components of velocity as well as the solid volume fraction for a node at the center of the bed over the 10 seconds time-averaging period for the 100 micron particles at 6 times the nominal minimum fluidization.

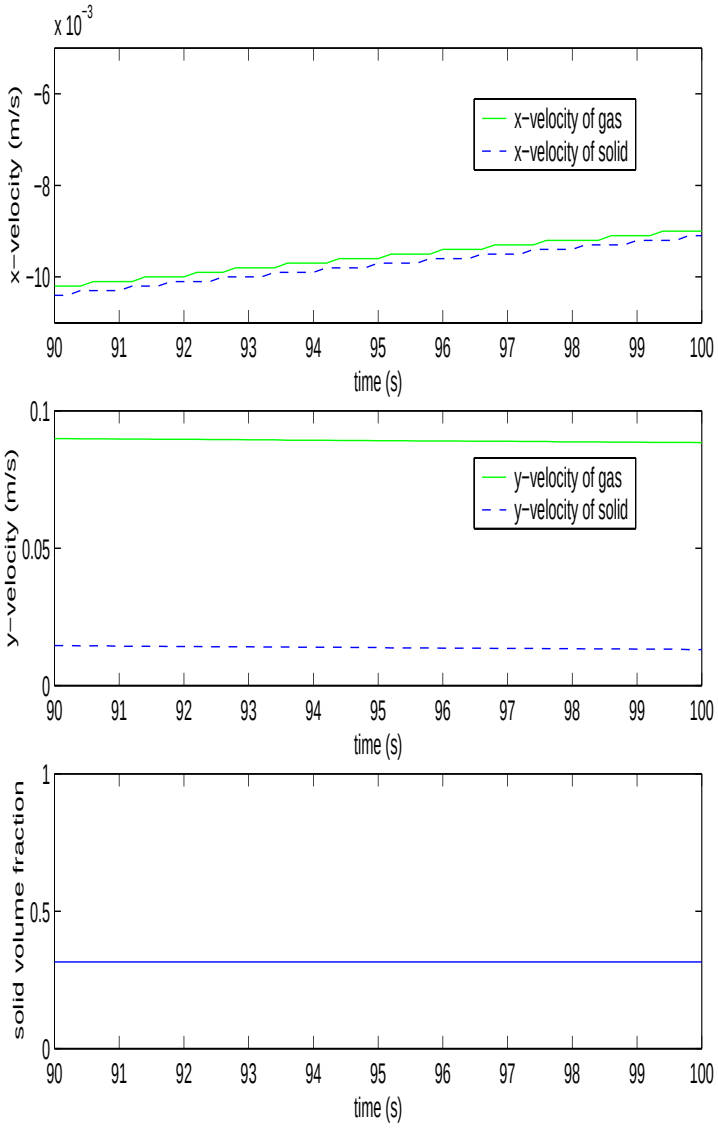


Figure 7.12: Time evolution of x and y-components of velocity as well as the solid volume fraction for a node at the center of the bed over the 10 seconds time-averaging period for the 125 micron particles at 6 times the nominal minimum fluidization.

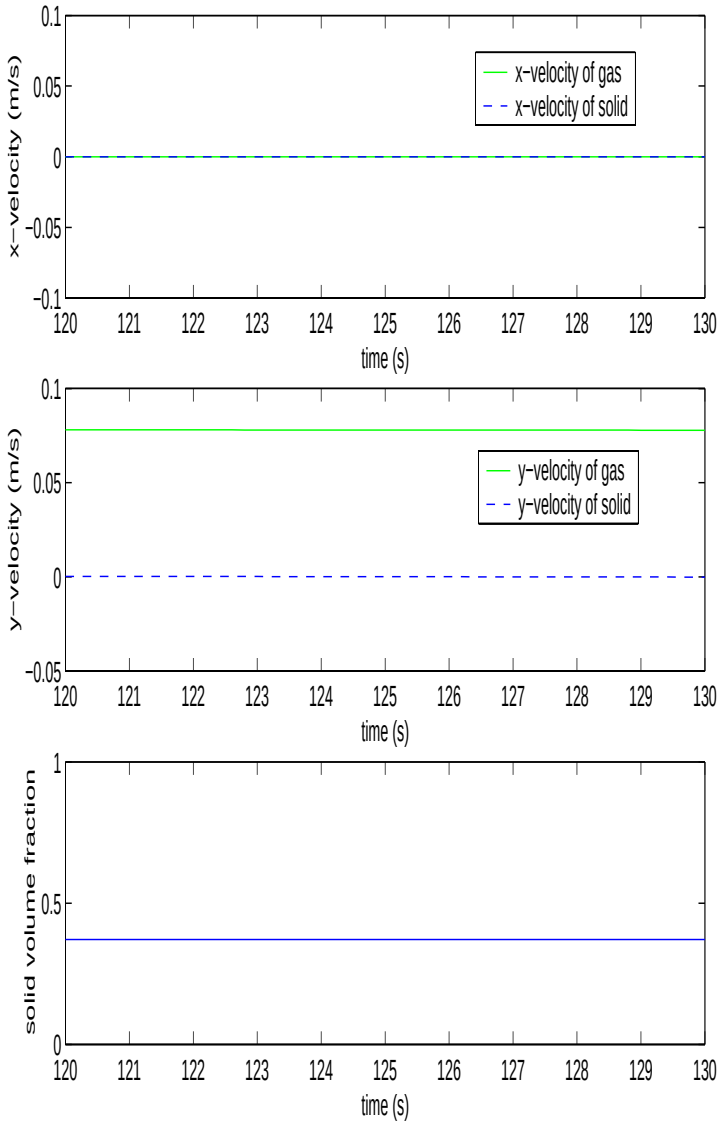


Figure 7.13: Time evolution of x and y-components of velocity as well as the solid volume fraction for a node at the center of the bed over the 10 seconds time-averaging period for the 150 micron particles at 4 times the nominal minimum fluidization.

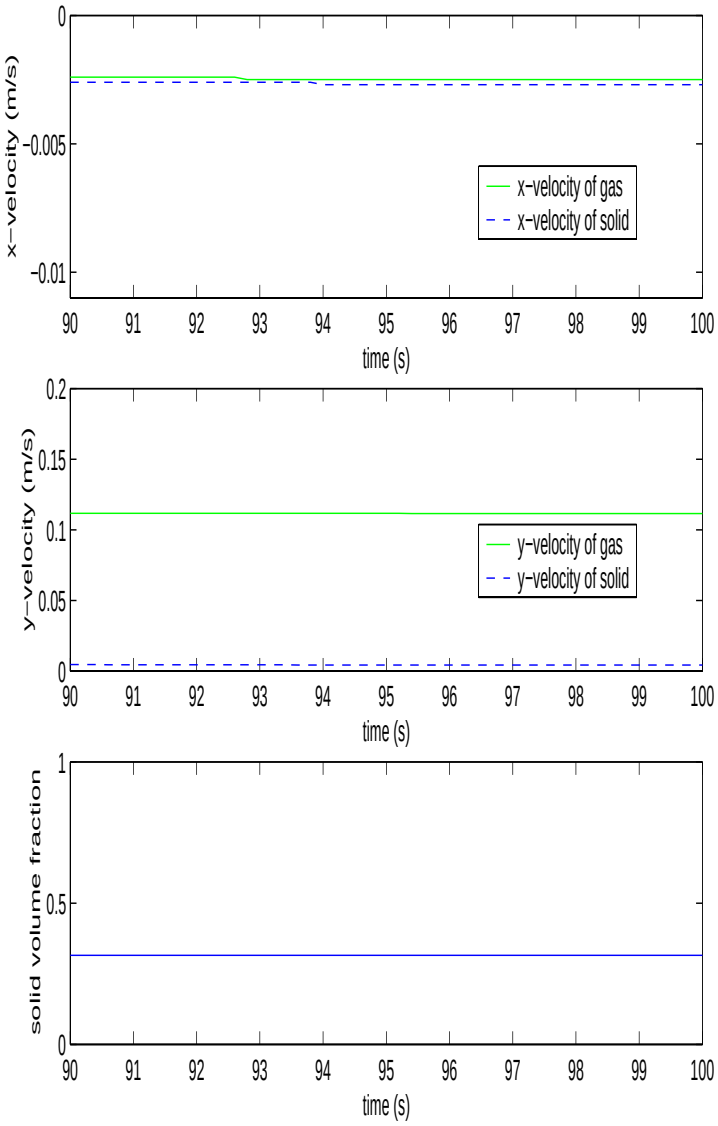


Figure 7.14: Time evolution of x and y-components of velocity as well as the solid volume fraction for a node at the center of the bed over the 10 seconds time-averaging period for the 150 micron particles at 6 times the nominal minimum fluidization.

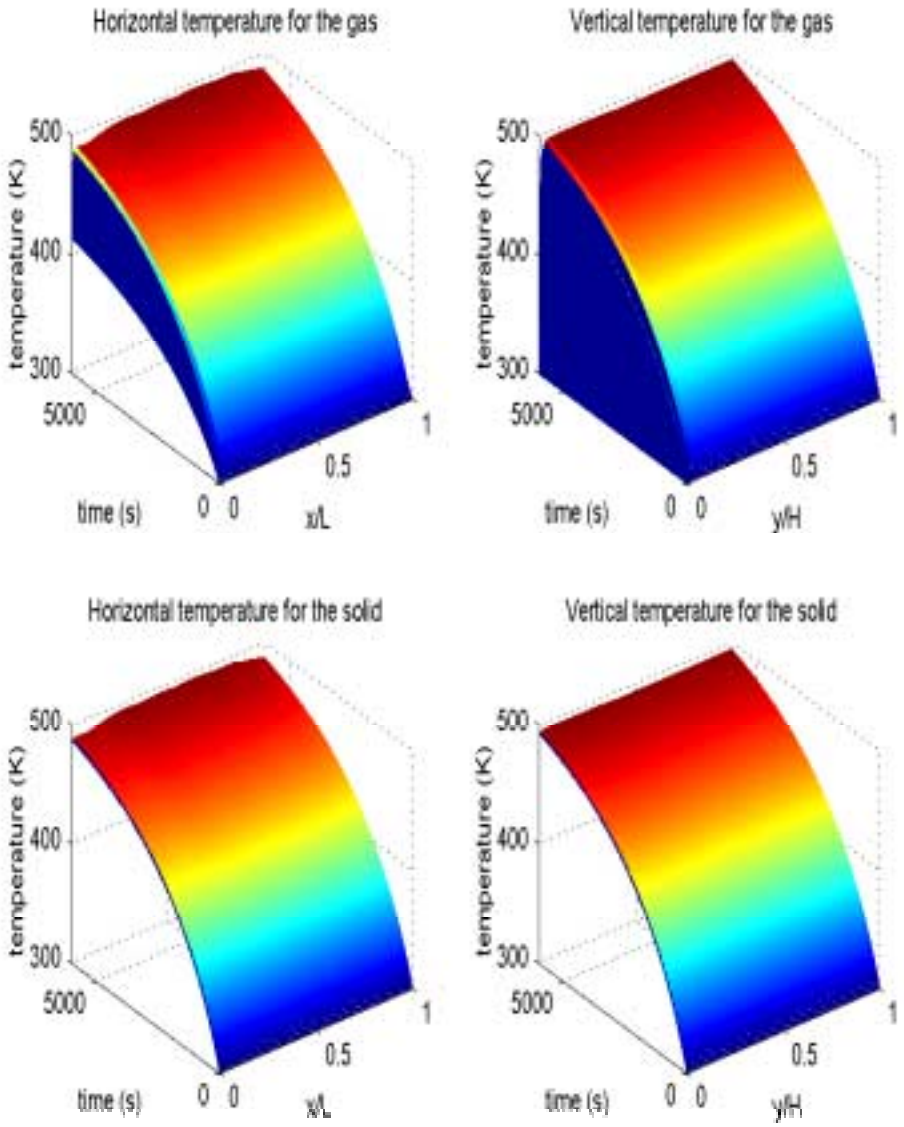


Figure 7.15: Time evolution of the temperature for the 100 micron particles at 6 times minimum fluidization.

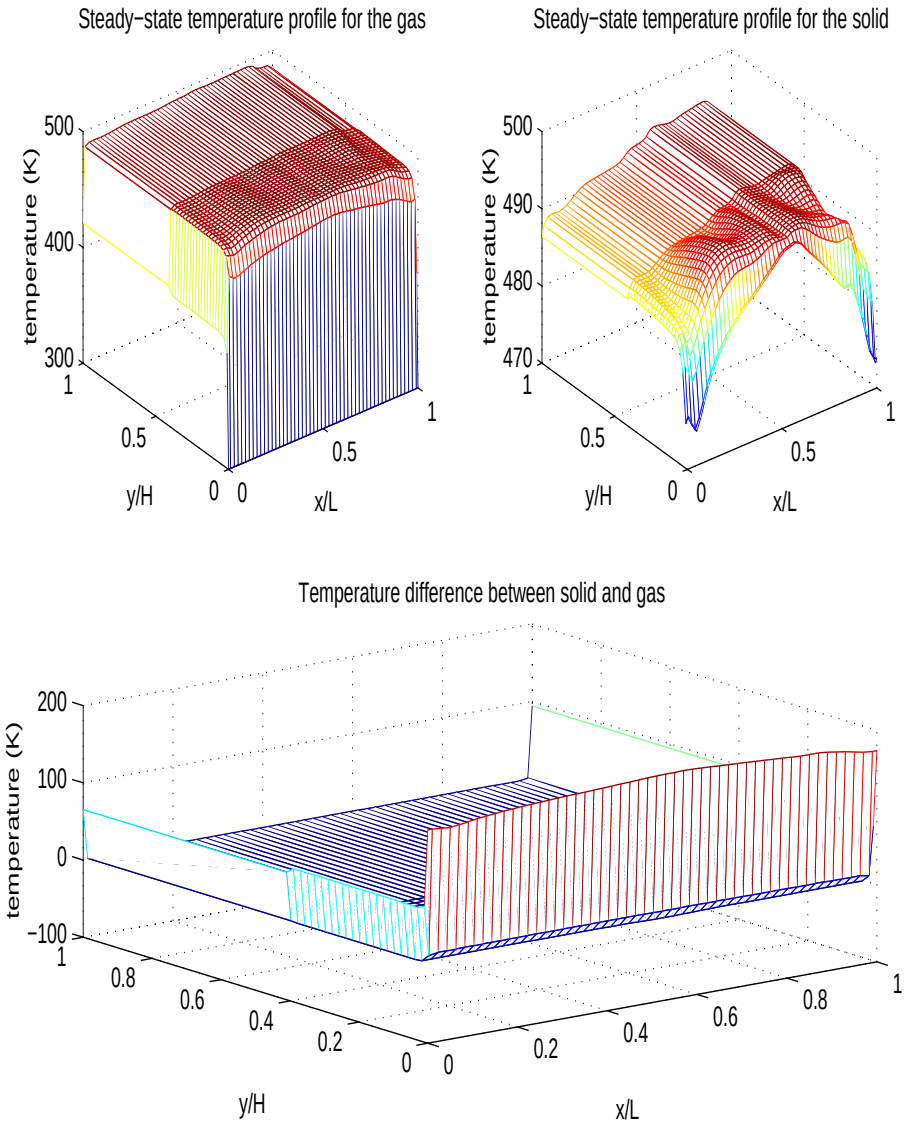


Figure 7.16: Steady-state temperature profiles for the 100 micron particles at 6 times minimum fluidization.

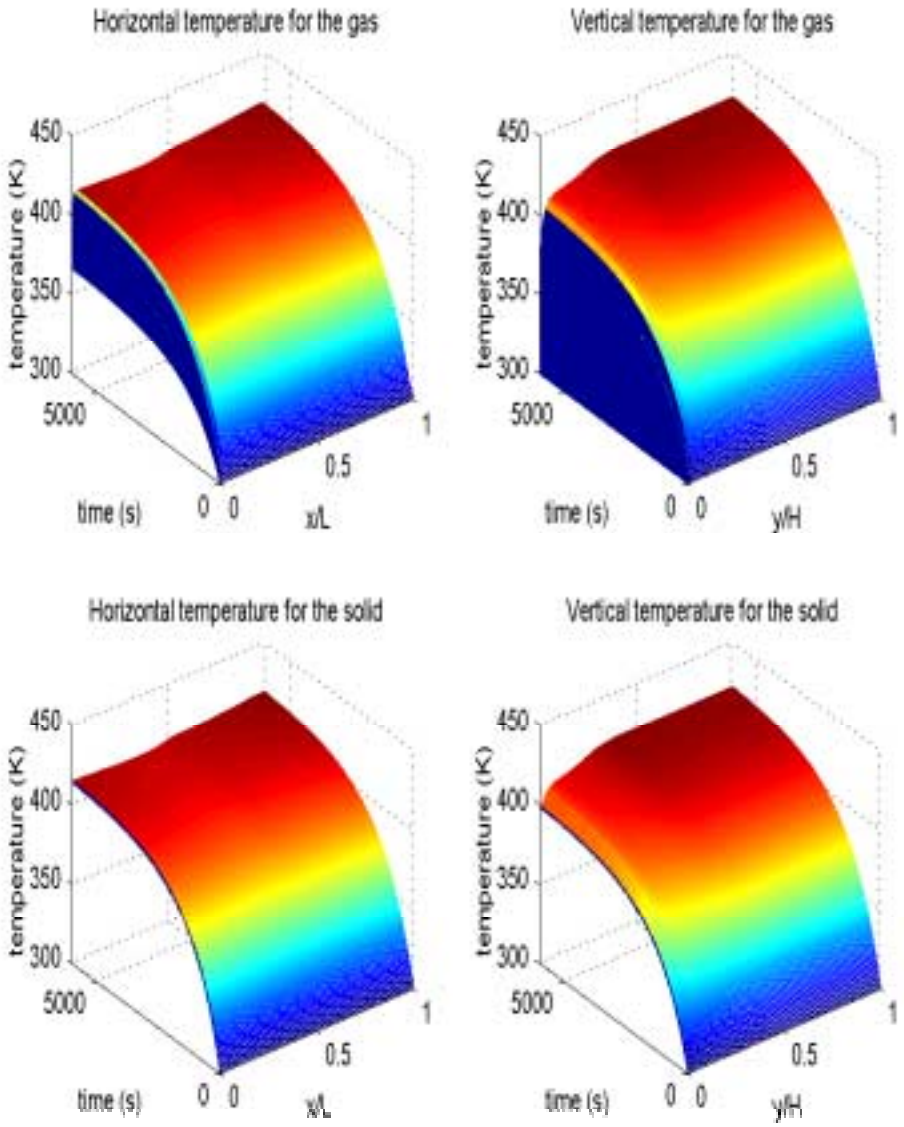


Figure 7.17: Time evolution of the temperature for the 125 micron particles at 6 times minimum fluidization.

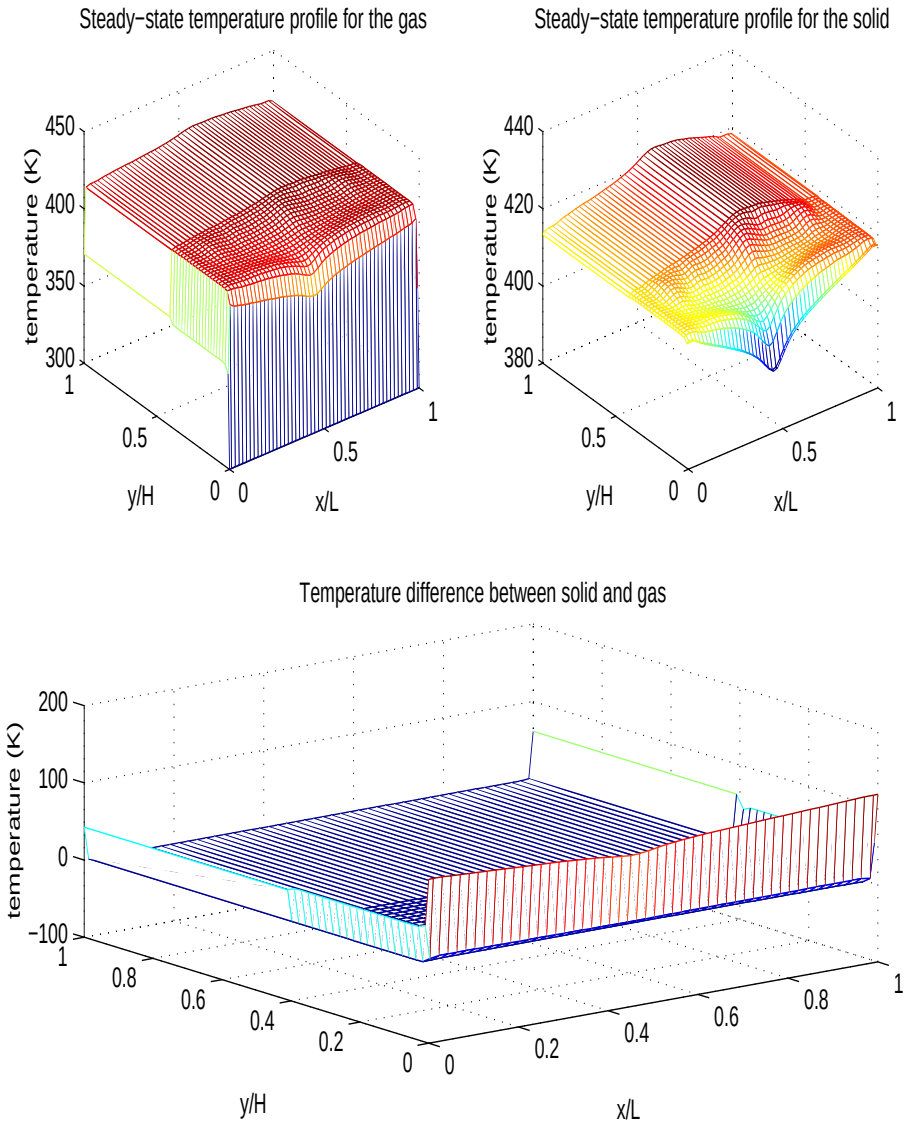


Figure 7.18: Steady-state temperature profiles for the 125 micron particles at 6 times minimum fluidization.

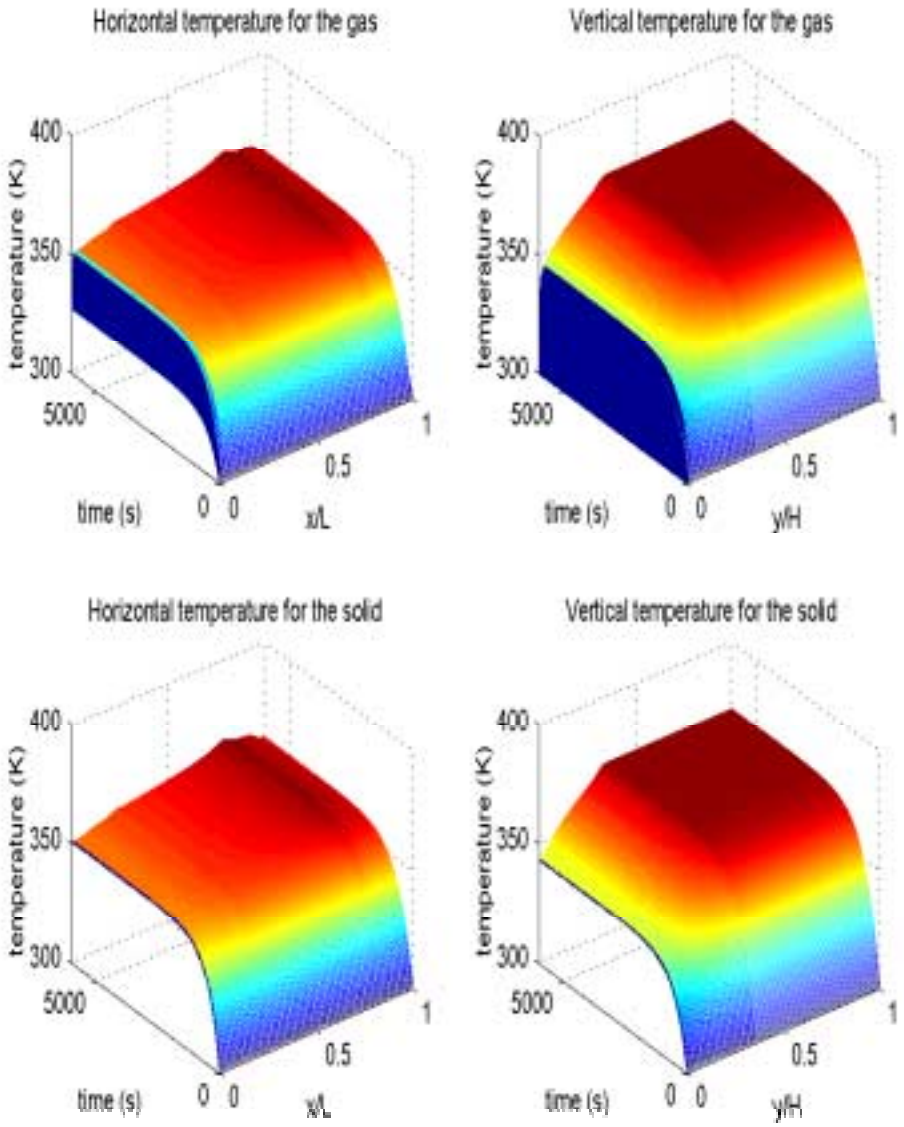


Figure 7.19: Time evolution of the temperature for the 150 micron particles at 4 times minimum fluidization.

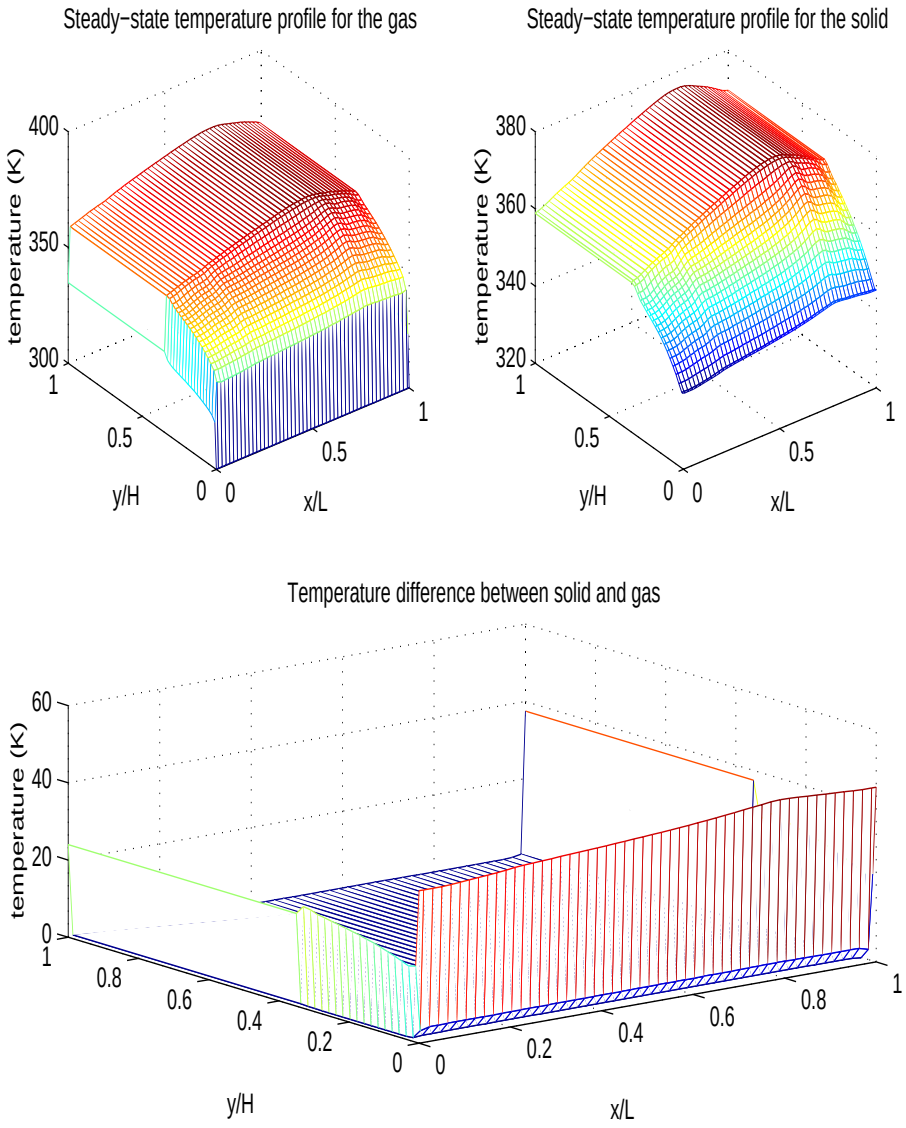


Figure 7.20: Steady-state temperature profiles for the 150 micron particles at 4 times minimum fluidization.

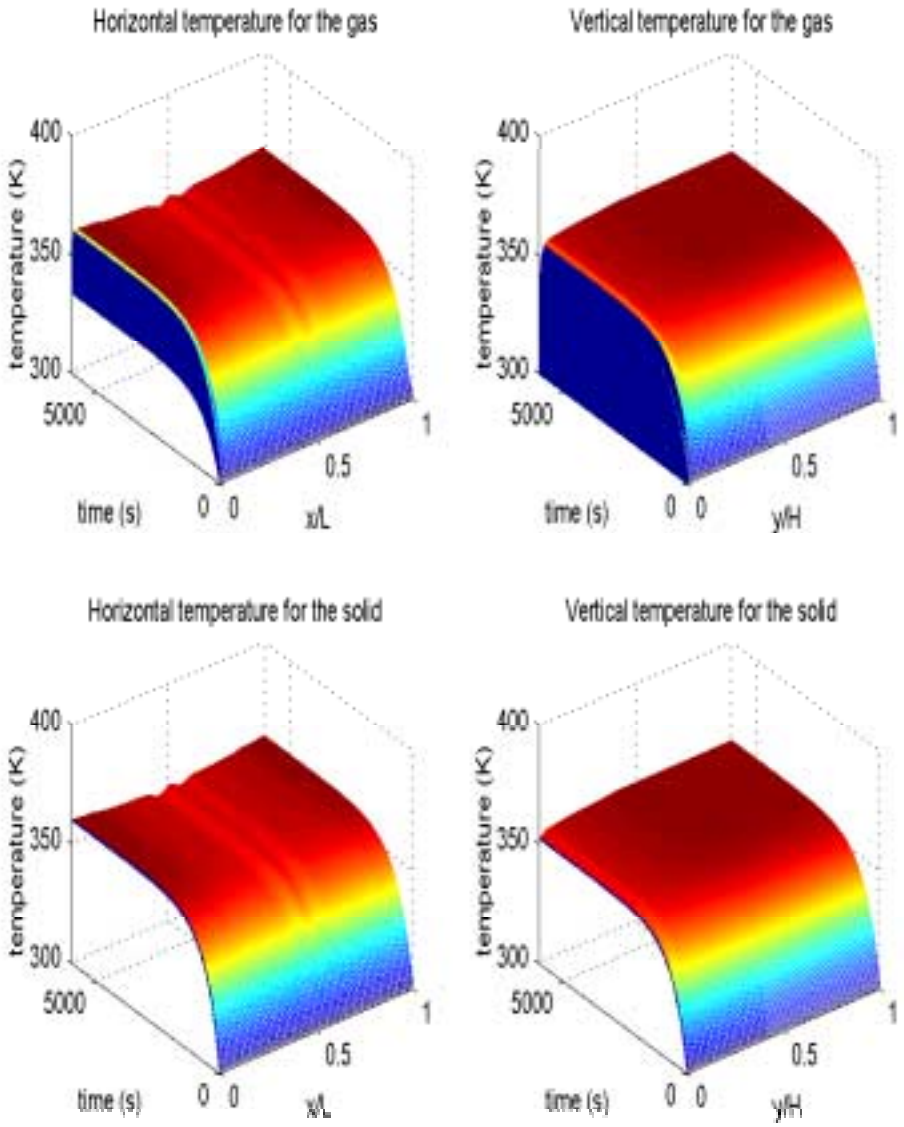


Figure 7.21: Time evolution of the temperature for the 150 micron particles at 6 times minimum fluidization.

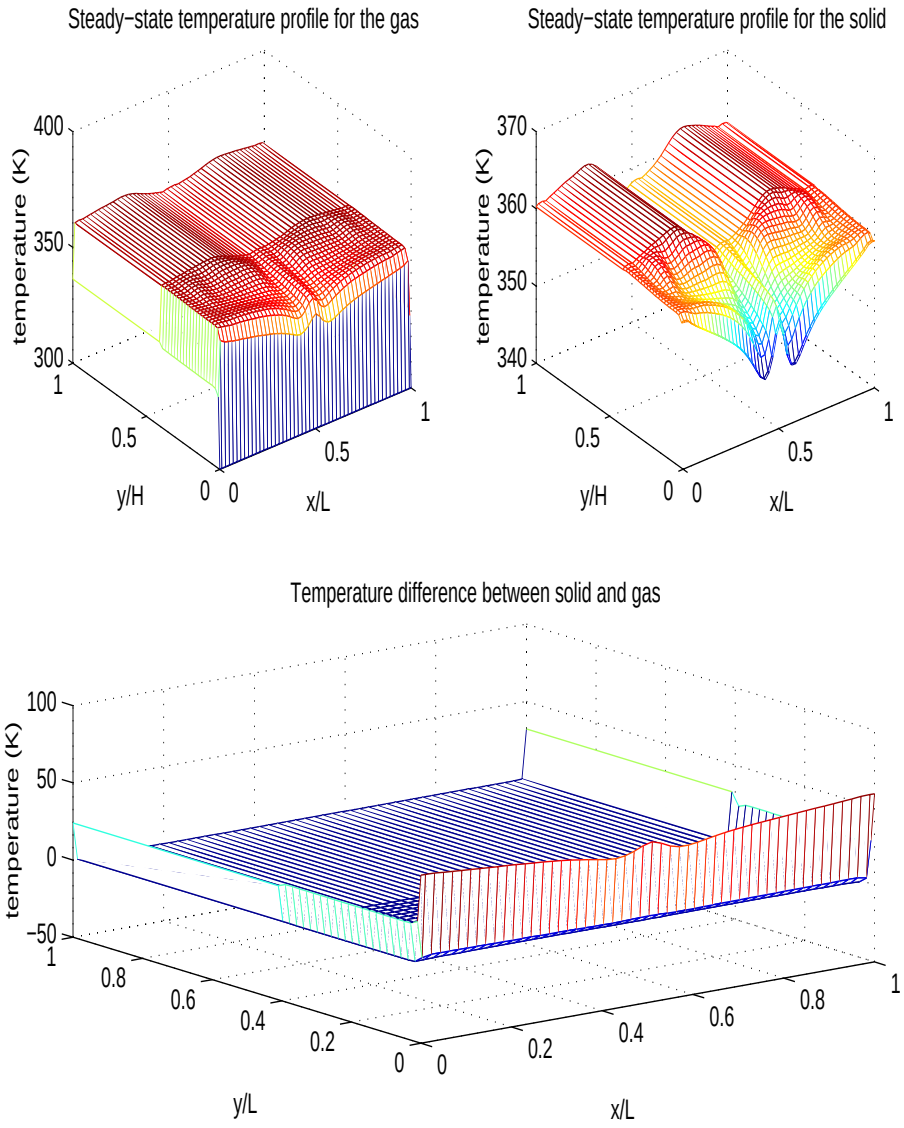


Figure 7.22: Steady-state temperature profiles for the 150 micron particles at 6 times minimum fluidization.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

From the preceding studies of the packed bed, we can conclude that the spatial temperature variations in the solid phase are negligible, i.e. the solids can be heated uniformly. This could be attributed to a combination of a uniform electric field along with the small conductive resistance induced by the size of the packed bed. Perhaps a lumped capacitance approach could be used to model the solid phase under these conditions. It is possible to produce a temperature difference of more than 5 degrees between the solid and the gas in about 15-25 percent of the packed bed. As expected, the electric field has greater influence on the temperature difference than the inlet gas velocity and particle diameter. An increase in particle diameter increases the temperature difference by a greater factor at nodes near the center than near the wall while an increase in the electric field increases the temperature difference by a greater factor at nodes near the wall than near the center. The inlet gas velocity effects the temperature difference at both nodes similarly. We also see that larger particles at higher inlet gas velocities and higher uniform electric fields produce the greatest temperature difference that penetrates the furthest into the packed bed. From the thermal perspective, a small packed bed is an efficient reactor.

From the preceding studies of the fluidized bed, we can conclude that homogeneous fluidization can be achieved for the particle diameters considered in this study

even though the stability criterion discussed in chapter 6 and the Geldart classification chart predict that bubbling should occur for such particles. In the current study we are dealing with a two-dimensional bed containing very low-density particles, but Geldart's data was obtained using three-dimensional cylindrical fluidized beds and the stability criterion was derived assuming a one-dimensional idealized bed configuration. Equation (6.1) predicts the same minimum fluidization velocity for all beds, but it has been demonstrated experimentally by Saxena et. al [63] that the minimum fluidization velocity is higher in two-dimensional beds than in three-dimensional beds. This is qualitatively consistent with current results. The 150 micron particles at 4 times the nominal minimum fluidization velocity exhibit a behaviour which is typical of beds at minimum fluidization in which the bed expands and the solid particles are nearly stationary. The range of inlet velocities for which the bed maintains a homogeneous state is very narrow for gas-solid systems. When the inlet velocity is increased to 8 times the nominal minimum fluidization velocity, homogeneity breaks down.

The steady-state temperature profiles are asymmetric and nonmonotonic for the gas and solid phases due to the erratic distribution of velocity. It is very difficult to produce a temperature difference with similar penetrations in a fluidized bed under a uniform electric field because the heat transfer coefficient between solid and gaseous phases is very large. The temperature difference is fairly sizable at the inlet because the gas temperature is fixed while the solid is heated continuously. At the bed boundaries, the gas looses energy at faster rate than the solid because the gas has a much lower thermal capacitance than the solid hence inducing a temperature difference between the two phases. The temperature difference between the gas and solid is very small within the bed even with a non-uniform electric field. A non-uniform electric field in the flow direction cannot overcome the effects of the large heat transfer coefficient between the gas and solid and the much lower thermal capacitance of the gas relative to the solid. Also, to achieve a sizable temperature difference around the bed perimeter, the current bed must be heated to temperatures of about 1500 K. Hence, from the thermal point of view the packed bed is a more efficient reactor than

a fluidized bed.

8.2 Future Work

Based on the conducted study, it is recommended that:

- A more detailed model of the electric field needs to be implemented. A simplified electric field model will not capture any potential "hot zones" that a model based on Maxwell's equations can capture.
- A mass transfer model needs to be implemented to observe the temperature effects on the concentration gradients and vice versa. This will allow for observation of product yields in a particular reaction.
- A new correlation for the gas-solid heat transfer coefficient accounting for microwave heating needs to be determined. The current correlation was obtained under conventional heating conditions.
- Extend the current model to account for bubbling and to investigate three-dimensional fluidized beds. This can be achieved as computational capabilities are enhanced.
- Investigate the effects of the electric field on the hydrodynamics. The electric field may influence the motion of the solids because of attraction/repulsion forces which may develop between closely interacting particles.
- Nonthermal effects such as mixing (fluidized beds), contact area, and gas residence time need to be incorporated into the models for predicting the efficiency of reactions. Mixing is very difficult to incorporate into any model, but a possible indicator of the extent of mixing, such as the granular temperature, can be used to accomplish this.
- Ultimately, an optimization of the conditions which will yield the best reaction rates needs to be performed, i.e. the optimal particle diameter, electric field, and inlet gas velocity, should be determined.

- To develop a molecular dynamics or a quantum mechanical model of an individual catalyst particle to demonstrate selective heating.

Bibliography

[1] G. Bond, R. B. Moyes, and D. A. Whan. *Catalysis Today*, Vol. 17, pp. 427-437, 1993.

[2] F. Chemat, D. C. Esveld, M. Poux, and J. L. DiMartino. *Journal of Microwave Power and Electromagnetic Energy*, Vol. 33, No. 2, 1998.

[3] H. C. Kim, H. Y. Kim, and S. I. Woo. *Journal of Chemical Engineering of Japan*, Vol. 32, No. 2, 1999.

[4] J. Berlan, P. Giboreau, S. Lefeuvre and C. Marchand. *Tetrahedron Letters*, 1991, Vol. 32, p. 2363.

[5] D. A.C Stuerga and P. Gaillard. *Journal of Microwave Power and Electromagnetic Energy*, Vol. 31, No. 2, 1996.

[6] G. Roussy and J. A. Pearce. *Foundations and Industrial Applications of Microwaves and Radio Frequency Fields*, John Wiley and Sons Ltd, West Sussex, England, 1995.

[7] D. Gidaspow. *Multiphase Flow and Fluidization, Continuum and Kinetic Theory Descriptions*, Academic Press, 1994.

[8] G. Nimtz, P. Marquart, and H. Gleiter. *Journal of Crystal Growth*, Vol. 86, 66, 1988.

[9] FLUENT Incorporated, 10 Cavendish Court, Centerra Park Lebanon, New Hampsure 03766.

[10] J. E. Lanz. *A Numerical Model of Thermal Effects in a Microwave Irradiated Catalyst Bed M.S. Thesis, Virginia Polytechnic Institute and State University*, Blacksburg, 1998.

[11] F. Faucher. *A Numerical Model of a Microwave Heated Fluidized Bed M.S. Thesis, Virginia Polytechnic Institute and State University*, Blacksburg, 1998.

[12] J. R. Thomas Jr. *Catalysis Letters*, Vol. 49, pp. 137-141, 1997.

[13] M. S. Ioffe, S. D. Pollington, and J. K.S. Wan. *Journal of Catalysis*, Vol. 151, pp. 349-355, 1995.

[14] G. Roussy, S. Hilaire, J. M. Thiebaut, G. Maire, S. Ringler, and F. Garin. *Applied Catalysis A: General*, Vol. 156, pp. 167-180, 1997.

[15] J. K.S. Wan. *Research on Chemical Intermediates*, Vol. 19, no. 2, pp. 147-158, 1993.

[16] S. L. Soo. *Fluid Dynamics of Multiphase Systems*, Blaisdell, Waltham, Mass. 1967.

[17] D. Butterworth and G.F.Hewitt. *Two-Phase Flow and Heat Transfer*, Oxford University Press, London, 1977.

[18] A. E. Green and P. M. Naghdi. *Arch. Rat. Mech. Anal.*, Vol. 24, p. 243, 1967.

[19] I. Muller. *Arch. Rat. Mech. Anal.*, Vol. 28, p. 1, 1968.

[20] T. Kawaguchi, T. Tanaka, and Y. Tsuji. *Powder Technology*, Vol. 96, pp. 129-138, 1998.

[21] M. Hassanizadeh and W. Gray. *Adv. Water Resources*, Vol. 2, p. 131, 1979.

[22] T. B. Anderson and R. Jackson. *Ind. Eng. Chem. Fund.*, Vol. 6, no. 4, 1967.

[23] D. A. Drew and L. A. Segel. *Stud. Appl. Math.*, Vol. 50, p. 233, 1971.

[24] M. N. Bogere. *Chemical Engineering Science*, Vol. 51, no. 4, pp. 603-622, 1996.

[25] J. Ding and D. Gidaspow. *AIChE Journal*, Vol. 36, no. 4, pp. 523-538, 1990.

[26] J. X. Bouillard, D. Gidaspow, and R. W. Lyczkowski. *Powder Technology*, Vol. 66, pp. 107-118, 1991.

[27] S. Benyahia, H. Arastoopour, and T. Knowlton. *Fluidization IX*, edited by L.S. Fan and T. Knowlton, 1998.

[28] S. Benyahia, H. Arastoopour, and T. M. Knowlton, and H. Massah. *Powder Technology,* Vol. 112, pp. 24-33, 2000.

[29] J. R. Thomas and F. Faucher. *Journal of Microwave Power and Electromagnetic Energy,* Vol. 35, no. 3, pp. 165-174, 2000.

[30] W. U. Choudhury. *Heat Transfer and Flow Characteristics of Conductive Porous Media with Energy Generation*, Ph.D. Dissertation, University of Wisconsin, 1968.

[31] E. Zahavi. *Int. J. Heat Mass Transfer*, Vol. 14, pp. 835-857, 1971.

[32] Z. H. Wang and G. Chen, Heat and Mass Transfer in Batch Fluidized-Bed Drying of Porous Particles, *Chemical Engineering Science,* Vol. 55, pp. 1857-1869, 2000.

[33] Z. H. Wang and G. Chen. *Ind. Eng. Chem. Res.*, Vol. 39, pp.775-782, 2000.

[34] T. Minoura, and Y. Sakamoto, T. Suzuki, and S. Toyama. *Society of Chemical Engineers Japan, Bunkyo Ku Tokyo*, Vol. 14, no. 5, pp. 583-592, 1988.

[35] G. Roussy, S. Jassam, and J. M. Thiebaut. *Journal of Microwave Power and Electromagnetic Energy*, Vol. 30, no.3, pp.178-187, 1995.

[36] F. J. Elvin. The Use of Dielectric Properties in the Design of a Reactor Heated by Microwaves. Coastal Technology Inc., Coastal Tower, 9 Greenway Plaza, Houston, TX 77046.

[37] E. Sizgek and G. D. Sizgek. Microwave Heated Fluidized Bed Calcination of Waste Containing Ceramic Powders. Australian Nuclear Science and Technology Organization, Private Mail Bag 1 Menai 2234, N.S.W., Australia.

[38] C. A. Balanis. *Advanced Engineering Electromagnetics*, Wiley, 1989.

[39] D. J. Griffiths. *Introduction to Electrodynamics*, Prentice Hall, 1999.

[40] M. Kaviany. *Principles of Heat Transfer in Porous Media*, Springer, 1995.

[41] A. Dybbs, and R. V. Edwards. *Fundamentals of Transport Phenomena in Porous Media*, Bear and Corapcioglu, 1984.

[42] J. Happel and H. Brenner. em Low Reynolds Number Hydrodynamics, Martinus Nijhoff Publishers, 1986.

[43] I. F. Macdonald, M. S. El-Sayed, K. Mow, and F. A.L. Dullien. *Ind. Eng. Chem. Fund.*, Vol. 18, pp. 199-208.

[44] S. Ergun. *Chem. Eng. Prog.*, Vol. 48, pp. 89-94.

[45] P. Cheng and H. Zhu. *Int. J. Heat and Mass Transfer*, Vol. 30, pp. 2373-2383, 1987.

[46] D. Vortmeyer and J. Schuster. *Chem. Eng. Sci.*, Vol. 38, pp. 1691-1699, 1983.

[47] J. P. Holman. *Heat Transfer*, McGraw Hill, 1972.

[48] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*, McGraw Hill, 1980.

[49] D. Kunii and O. Levenspiel. em Fluidization Engineering, Butterworth and Heinemann, 1991.

[50] D. Geldart. *Powder Technology*, Vol. 7, p. 285, 1973.

[51] D. Geldart, and A. R. Abrahamsen. *Powder Technology*, Vol. 19, pp. 133-136, 1978.

[52] L. G. Gibilaro. *Fluidization-Dynamics*, Butterworth and Heinemann, 2001.

[53] W. K. Lewis, E. R. Gilliland, and W. C. Bauer. *Ind. Eng. Chem.*, Vol. 41, p. 1104, 1949.

[54] A. K. Kothari. *Analysis of Fluid-Solid Heat Transfer Coefficients in Fluidized Beds*, M.S. Thesis, Illinois Institute of Technology, 1967.

[55] D. J. Gunn. *International Journal of Heat and Mass Transfer*, Vol. 21, p. 467, 1978.

[56] J. S. Walton, R. L. Olson, and O. Levenspiel. *Ind. Eng. Chem.*, Vol. 44, 1952.

[57] N. I. Siriomiatnikov and V. F. Volkov. *Protzessi v Kipyaschem Sloye*, Metalurgizdat, Sverdlovsk, 1959.

[58] FLUENT 4.4 User's Guide Vol. 2, pp. 9-18 - 9-24, FLUENT INCORPORATED, 1984-1997.

[59] B. Carnahan, H. A. Hunter, and J. O. Wilkes. *Applied Numerical Methods*, Wiley, 1969.

[60] S. Benyahia, H. Arastoopour, and T. Knowlton. *Fluidization IX*, 1998.

[61] I. K. Gamwo, Y. Soong, R. W. Lyczkowski. *Powder Technology*, Vol. 103, pp. 117-129, 1999.

[62] S. Benyahia, H. Arastoopour, and T. M. Knowlton. *Powder Technology*, Vol. 112, pp. 24-33, 2000.

[63] W. Y. Wu, S. C. Saxena, R. L. Trojnarski, and M. V. Morris. *Energy*, Vol. 21, no. 10, pp. 825-833, 1996.

Appendix A

Packed Bed Program

```
c    This program will determine the temperature and velocity profiles and
c    the temperature difference between the solid and gas in a small
c    cylindrical packed bed.
c
c    The following is a definition of the input variables.
c
c     initer = number of inner iterations.
c     ncr = number of cells in the radial direction.
c     ncz = number of cells in the axial direction.
c     nnf = total number of nodes for a single phase.
c     nnt = total number of nodes for both phases.
c     nn2 = total number of nodes in the computation including the fluids.
c     ntime = number of time steps.
c     job = parameter set to zero to solve Ax = b in the matrix solver.
c     kg = conductivity of glass (W/m\cdot K).
c     hn = heat transfer coefficient between container and surroundings
c          (W/$m^2$\cdot K).
c     rb = radius of the bed (m).
c     ro = radius of the bed plus that of the container (m).
c     rc = radius of the waveguide (m).
```

```
c     area = cross-sectional area of the bed (m).
c     h = height of the bed (m).
c     dr = length of a control volume in the computational grid (m).
c     dz = height of a control volume in the computational grid (m).
c     dt = time step (s).
c     epsi = volume fraction of the bed at the center.
c     dpt = particle diameter of the metallic catalyst particle (m).
c     pf = volume fraction of metallic catalyst particle in the ceramic
c          support.
c     rhoal = density of alumina (kg/$m^3$).
c     rhopt = density of platinum (kg/$m^3$).
c     vin = inlet gas velocity (m/s).
c     vflow = inlet volume flow rate of the gas ($m^3$/s).
c     dp = particle diameter (m).
c     rhos = density of the ceramic support/metallic catalyst mixture
c            (kg/$m^3$).
c     tmpin = inlet gas temperature (K).
c     frequ = microwave frequency (Hz).
c     eps0 = permitivity of free space (f/m).
c     mu0 = permeability of free space (h/m).
c     c0 = speed of light (m/s).
c     ef0 = electric field within the bed (V/m).
c     dpdz = pressure drop across the bed (N/$m^3$).
c     tol = tolerance in the mass flow iterations.
c     tol1 = tolerance in velocity iterations.
c     bc = constant for bessel function J0.
c
c     flag = determines whether the support will be alpha or gamma-alumina:
c            if set to 1, then alpha alumina is used; otherwise gamma alumina
c            is used.
c
c     flag1 = controls electric field: if set to 1, the electric field is
```

```
c             uniform; if set to 2, the electric field varies linearly in
c             the flow direction; otherwise, the electric field is described
c             by a radial bessel function.
c
c     flag2 = determines the inlet boundary condition for the solid phase:
c             if set to 1, the solid is insulated; otherwise the solid is
c             set to the inlet temperature of the gas.
c
      use msimsl
      implicit real*8(a-h,o-y)
real*8 kg,mu0,z
      integer ncr,ncz,nnf,nnt,ntime,flag,flag2,flag1,nn,lda,ipvt,nout
integer job,flag3
c
      parameter(initer=4)
      parameter(ncr=30)
      parameter(ncz=20)
      parameter(nnf=2*(ncz+ncr)+ncr*ncz)
      parameter(nnt=2*nnf)
parameter(nn2=nnt+ncr+2)
      parameter(nn=nn2)
      parameter(lda=nn)
c
      dimension tprev(nnt),z(nn),ipvt(nn),temp(nnt),a(lda,nn),rhs(nn2)
dimension taves(ncz),efelds(ncz),efs(nnf),tavesf(ncr)
dimension vguess(ncr+1)
c
common/param/flag,flag2,flag3
common/dielectric/pi,frequ,eps0,c0,mu0,ef0
common/solgas/rhos,dp,pf,dpt,vf,tmpin,kg,epsi,rb,ro,hn
common/delta/h,dt,dr,dz
c
```

```
open(1,file='c:\pabed\radiuspb')
      open(2,file='c:\pabed\heightpb')
      open(3,file='c:\pabed\timepb')
      open(4,file='c:\pabed\tfrpb500')
      open(5,file='c:\pabed\tsrpb500')
      open(6,file='c:\pabed\tfzpb500')
      open(7,file='c:\pabed\tszpb500')
open(8,file='c:\\pabed\velpb500')
open(9,file='c:\pabed\tfbedss500')
open(10,file='c:\pabed\tsbedss500')
open(11,file='c:\pabed\tdiff500')
c
      ntime=21
      pi=dacos(-1.d0)
      job=0
kg=1.4d0
hn=2.d0
rb=1.d-2
ro=1.1d-2
rc=6.d-1
area=pi*rb**2/2.d0
h=2.5d-2
dr=rb/dble(ncr)
dz=h/dble(ncz)
      dt=1.d2
epsi=4.d-1
      dpt=4.d-8
      pf=.01d0
      rhoal=700.d0
      rhopt=21450.d0
vin=1.d-1
vflow=vin*area
```

```
      dp=5.d-4
      vf=(pf/rhopt)/((pf/rhopt)+(1.d0-pf)/rhoal)
rhos=pf*rhopt+(1.d0-pf)*rhoal
      tmpin=300.15d0
      frequ=.915d9
      eps0=8.854d-12
      mu0=4.d-7*pi
c0=3.d8
ef0=5.d3
flag=2
flag1=1
flag2=1
c flag3=2
efelds(ncz)=ef0
dpdz=-1.5d2
tol=1.d-6
tol1=1.d-4
bc=2.4049d0
c
      do i=1,nnt
       tprev(i)=tmpin
 temp(i)=tmpin
 rhs(i)=0.d0
      enddo
c
      do i=1,ncr+1
 rhs(nnt+i)=vin
 vguess(i)=vin
 if(i .lt. ncr+1) then
  tavesf(i)=tmpin
 endif
enddo
```

```
c
      do i2=1,ntime
       k=nnf+ncz+2
       l=nnt-2*ncz
       do i=1,ncz
  sum=0.d0
        do j=k,l,ncz+2
   sum=sum+temp(j)
  enddo
  taves(i)=sum/dble(ncr)
  k=k+1
  l=l+1
 enddo
c
       if(flag1 .eq. 1) then
        call efields(taves,efelds,eppm,ncz)
c   write(*,*) eppm
  do i=1,nnf-2*(ncz+ncr)
   efs(i)=ef0
  enddo
 elseif(flag1 .eq. 2) then
  call efields(taves,efelds,eppm,ncz)
  k=ncz+2
        l=nnf-2*ncz
        do i=1,ncz
         do j=k,l,ncz+2
     efs(j)=efelds(i)
   enddo
   k=k+1
   l=l+1
  enddo
 else
```

```
  k=ncz+2
  l=k+ncz-1
  p=1.d0
  do i=1,ncr
   r=p*dr
   do j=k,l
    efs(j)=ef0*dbsj0(bc*r/rc)
    call umach(2,nout)
   enddo
   k=k+ncz+2
   l=l+ncz+2
   p=p+1.d0
  enddo
 endif
c
       do k=1,nnt
        temp(k)=tprev(k)
       enddo
c
    5  do j2=1,initer
c
        call assemble(tprev,a,rhs,temp,efs,eppm,lda,nn,ncr,
1               ncz,nnf,nnt,nn2,vguess,tavesf,dpdz)
c    if((i2.eq.1) .and. (j2.eq.1)) then
c     do j3=1,lda
c      do j4=1,nn
c       write(1,*) j3, j4, a(j3,j4), rhs(j3)
c      enddo
c     enddo
c    endif
  call dgeco(a,lda,nn,ipvt,rcond,z)
c    write(*,*) 'the condition # is', 1.d0/rcond, i2
```

```
  call dgesl(a,lda,nn,ipvt,rhs,job)
c    if((i2.eq.ntime) .and. (j2.eq.3)) then
c        do j3=1,nn2
c        write(*,*) j3, rhs(j3)
c     enddo
c         endif
c
        do l=1,nnt
         temp(l)=rhs(l)
        enddo
  do i3=1,lda
   do j3=1,nn
    a(i3,j3)=0.d0
   enddo
  enddo
       enddo
c
       k=ncz+2
 l=2*ncz+1
 do i=1,ncr
  sum=0.d0
  do j=k,l
   sum=sum+temp(j)
        enddo
  tavesf(i)=sum/dble(ncz)
  k=k+ncz+2
  l=l+ncz+2
 enddo
c
 p=0.d0
 rhs1=0.d0
 do j2=nnt+2,nnt+ncr+1
```

```
  rhs1=rhs1+pi*((p+1.d0)**2-p**2)*dr**2*rhs(j2)/2.d0
  p=p+1.d0
 enddo
c write(*,*) rhs1, vflow
c
       if((dabs(vflow-rhs1).gt.tol).and.((vflow-rhs1).gt.0.d0))then
  dpdz=dpdz+dpdz/2.d0
  goto 5
 elseif((dabs(vflow-rhs1).gt.tol).and.((vflow-rhs1).lt.0.d0))then
  dpdz=dpdz-dpdz/2.d0
  goto 5
 endif
c
        vdiff=0.d0
  do j2=1,ncr+1
   if(dabs(rhs(nnt+j2)-vguess(j2)) .gt. vdiff) then
    vdiff=dabs(rhs(nnt+j2)-vguess(j2))
   endif
  enddo
  if(vdiff .gt. tol1) then
   do j2=1,ncr+1
    vguess(j2)=rhs(nnt+j2)
   enddo
   goto 5
  endif
c
c       if(mod(i2,10) .eq. 1) then
        write(4,11) rhs(int(ncz/2)),(rhs(l),l=int(ncz/2)+ncz+1,(ncr-1)*
1              (ncz+2)+int(ncz/2)+ncz+1,ncz+2),rhs((ncr-1)*
1              (ncz+2)+int(ncz/2)+2*(ncz+1))
        write(5,11) rhs(nnf+int(ncz/2)),(rhs(l),l=nnf+int(ncz/2)+ncz+1,
1    nnf+(ncr-1)*(ncz+2)+int(ncz/2)+ncz+1,ncz+2),rhs(nnf+
```

```
     1    (ncr-1)*(ncz+2)+int(ncz/2)+2*(ncz+1))
        write(6,15) (rhs(l),l=int(ncr/2)*(ncz+2)+ncz+1,int(ncr/2)*(ncz+
1               2)+2*(ncz+1))
        write(7,15) (rhs(l),l=nnf+int(ncr/2)*(ncz+2)+ncz+1,nnf+int(ncr/
1               2)*(ncz+2)+2*(ncz+1))
c
   11   format(1x,32(f6.2,3x))
   15   format(1x,22(f6.2,3x))
   16   format(f7.1,2x,f7.1,2x,f7.1,2x,f7.1,2x,f7.1,2x,f7.1)
c       endif
       if(i2 .eq. 1) then
  do l=1,ncr+2
   if(l .eq. 1) then
    write(1,17) 0.d0
   elseif(l .eq. ncr+2) then
    write(1,17) 1.d0
   else
          write(1,17) (dble(l)-1.5d0)*dr/rb
   endif
  enddo
  h=2.5d-2
       do l=1,ncz+2
   if(l .eq. 1) then
    write(2,17) 0.d0
   elseif(l .eq. ncz+2) then
    write(2,17) 1.d0
   else
          write(2,17) (dble(l)-1.5d0)*dz/h
   endif
       enddo
   17   format(f6.4)
       endif
```

```
c
c       if(mod(i2,10) .eq. 1) then
        write(3,18) dble(i2)*dt
   18   format(f7.1)
c       endif
c
        if(i2 .lt. ntime) then
         do l=1,nnt
          tprev(l)=rhs(l)
         enddo
        endif
       enddo
c
      k=0
      do l=1,ncz+2
 if(l .eq. 1) then
  write(9,20) (rhs(m), m=ncz+1,nnf-2*ncz-1,ncz+2)
 elseif(l .eq. ncz+2) then
  write(9,20) (rhs(m), m=2*(ncz+1),nnf-ncz,ncz+2)
 else
  k=k+1
  write(9,19) rhs(k),(rhs(m),m=ncz+1+k,nnf-2*ncz-1+k,ncz+2),
1      rhs(nnf-ncz+k)
 endif
enddo
c
k=0
      do l=1,ncz+2
 if(l .eq. 1) then
  write(10,20) (rhs(m), m=nnf+ncz+1,2*nnf-2*ncz-1,ncz+2)
 elseif(l .eq. ncz+2) then
  write(10,20) (rhs(m), m=nnf+2*(ncz+1),2*nnf-ncz,ncz+2)
```

```
 else
  k=k+1
  write(10,19) rhs(nnf+k),(rhs(m),m=nnf+ncz+1+k,2*nnf-2*ncz-1+
1              k,ncz+2), rhs(2*nnf-ncz+k)
   endif
      enddo
c
      k=0
      do l=1,ncz+2
 if(l .eq. 1) then
  write(11,22) (rhs(m+nnf)-rhs(m), m=ncz+1,nnf-2*ncz-1,ncz+2)
 elseif(l .eq. ncz+2) then
  write(11,22) (rhs(m+nnf)-rhs(m), m=2*(ncz+1),nnf-ncz,ncz+2)
 else
  k=k+1
  write(11,21) rhs(k+nnf)-rhs(k),(rhs(m+nnf)-rhs(m),m=ncz+1+k,
1             nnf-2*ncz-1+k,ncz+2), rhs(2*nnf-ncz+k)-rhs(nnf-ncz+k)
 endif
enddo
c
write(8,23) (rhs(l),l=nnt+1,nn2)
c
   19 format(1x,32(f6.2,3x))
   20 format(1x,30(f6.2,3x))
   21 format(1x,32(f9.4,3x))
   22 format(1x,30(f9.4,3x))
   23 format(1x,32(f6.4,3x))
write(*,*) 'the pressure drop is', dpdz
c
       close(1)
       close(2)
       close(3)
```

```
       close(4)
       close(5)
       close(7)
       close(8)
 close(9)
 close(10)
 close(11)
      end
c
      subroutine assemble(tprev,a,rhs,temp,efs,eppm,lda,nn,
1                   ncr,ncz,nnf,nnt,nn2,vguess,tavesf,dpdz)
      implicit real*8(a-h,o-y)
      real*8 kg,mug,mu0,nu
      integer flag,flag2,flag3,e,s,w,n,nn,lda,ncr,ncz,nnf,nnt,nn2
c
      dimension a(lda,nn),temp(nnt),tprev(nnt),rhs(nn2),tavesf(ncr)
      dimension efs(nnf),vguess(ncr+1)
c
common/param/flag,flag2,flag3
common/dielectric/pi,frequ,eps0,c0,mu0,ef0
common/solgas/rhos,dp,pf,dpt,vf,tmpin,kg,epsi,rb,ro,hn
common/delta/h,dt,dr,dz
c gas boundary nodes
do i=1,ncz
 tp=temp(i)
 a(i,i)=1.d0
 a(i,i+ncz+1)=-1.d0
 rhs(i)=0.d0
 k=nnf-ncz+i
 tp=temp(k)
 rtot=rb*dlog(ro/rb)/kg+rb/(hn*ro)
 a(k,k)=1.d0/rtot+eps(rb)*cgas(tp)/dr
```

```
 a(k,k-ncz-1)=-eps(rb)*cgas(tp)/dr
 rhs(k)=tmpin/rtot
      enddo
c gas inlet temperature boundary
      do i=ncz+1,nnf-2*ncz-1,ncz+2
 a(i,i)=1.d0
 rhs(i)=tmpin
enddo
c gas boundary nodes
      do i=2*(ncz+1),nnf-ncz,ncz+2
 a(i,i)=1.d0
 a(i,i-1)=-1.d0
 rhs(i)=0.d0
enddo
c gas eqns. for a typical node
j=ncz+2
m=2*ncz+1
p=0.d0
do i=1,ncr
   do i1=j,m
  n=i1+1
  s=i1-1
  w=i1-ncz-2
  e=i1+ncz+2
  j1=i1+nnf
  drw=dr
  dre=dr
  dzs=dz
  dzn=dz
  if(i .eq. 1) then
   w=i1-ncz-1
   drw=dr/2.d0
```

```
elseif(i .eq. ncr) then
 e=i1+ncz+1
 dre=dr/2.d0
endif
if(i1 .eq. j) then
 dzs=dz/2.d0
elseif(i1 .eq. m) then
 dzn=dz/2.d0
endif
      tp=temp(i1)
  te=temp(e)
  tw=temp(w)
tn=temp(n)
ts=temp(s)
tm=temp(j1)
ce=(cgas(tp)+cgas(te))/2.d0
cw=(cgas(tp)+cgas(tw))/2.d0
cn=(cgas(tp)+cgas(tn))/2.d0
cs=(cgas(tp)+cgas(ts))/2.d0
cpg=cpgas(tp)
ta=tavesf(i)
rhog=rhof(ta)
mug=viscf(ta)
re=dp*rhog*rhs(nnt+i+1)/mug
      pr=cpg*mug/cgas(tp)
      nu=2.d0+1.2d0*re**(0.5d0)*pr**(1.d0/3.d0)
      h1=nu*cgas(tp)/dp
r=(p+5.d-1)*dr
      aa=6.d0*(1.d0-eps(r))/dp
      h1=h1*aa
r=(p+1.d0)*dr
a(i1,e)=-eps(r)*ce*(p+1.d0)*dr*dz/dre
```

```
  r=p*dr
  a(i1,w)=-eps(r)*cw*p*dr*dz/drw
  r=(p+5.d-1)*dr
  a(i1,n)=-cn*eps(r)/dzn*((p+1.d0)**2-p**2)/2.d0*dr**2
  a(i1,s)=(-rhog*cpg*eps(r)*rhs(nnt+i+1)*((p+1.d0)**2-p**2)/2.d0*
1          dr**2-cs*eps(r)/dzs*((p+1.d0)**2-p**2)/2.d0*dr**2)
  a(i1,j1)=-h1*dz*((p+1.d0)**2-p**2)/2.d0*dr**2
  a(i1,i1)=rhog*cpg*eps(r)*dz*((p+1.d0)**2-p**2)/2.d0*dr**2/dt-
1           a(i1,e)-a(i1,w)-a(i1,s)-a(i1,j1)-a(i1,n)
  rhs(i1)=eps(r)*rhog*cpg*dz*((p+1.d0)**2-p**2)/2.d0*dr**2/dt*
1          tprev(i1)
 enddo
 j=j+ncz+2
 m=m+ncz+2
 p=p+1.d0
enddo
c solid boundary nodes
do i=nnf+1,nnf+ncz
 tp=temp(i)
   a(i,i)=1.d0
 a(i,i+ncz+1)=-1.d0
 rhs(i)=0.d0
 k=nnt-ncz+i-nnf
 tp=temp(k)
 rtot=rb*dlog(ro/rb)/kg+rb/(hn*ro)
 a(k,k)=1.d0/rtot+(1.d0-eps(rb))*cons(tp,pf)/dr
 a(k,k-ncz-1)=-(1.d0-eps(rb))*cons(tp,pf)/dr
 rhs(k)=tmpin/rtot
      enddo
c
if(flag2 .eq. 1) then
c insulated boundary
```

```
       do i=nnf+ncz+1,nnt-2*ncz-1,ncz+2
c   if(flag3 .eq. 2) then
   a(i,i)=1.d0
   a(i,i+1)=-1.d0
   rhs(i)=0.d0
c   else
c    r=0.d0
c    tp=temp(i)
c    tm=temp(i-nnf)
c    h2=2.d0*cgas(tm)/dp
c    a(i,i)=h2*dr-(1.d0-eps(r))*cons(tp,pf)
c    a(i,i+1)=(1.d0-eps(r))*cons(tp,pf)
c    rhs(i)=h2*dr*tm
c   endif
 enddo
else
c temperature boundary
 do i=nnf+ncz+1,nnt-2*ncz-1,ncz+2
        a(i,i)=1.d0
  rhs(i)=tmpin
 enddo
endif
c solid boundary nodes
do i=nnf+2*(ncz+1),nnt-ncz,ncz+2
 if(flag3 .eq. 2) then
  a(i,i)=1.d0
  a(i,i-1)=-1.d0
  rhs(i)=0.d0
 else
  r=rb
  tp=temp(i)
  tm=temp(i-nnf)
```

```
  h2=2.d0*cgas(tm)/dp
  a(i,i)=h2*dr-(1.d0-eps(r))*cons(tp,pf)
  a(i,i-1)=(1.d0-eps(r))*cons(tp,pf)
  rhs(i)=h2*dr*tm
 endif
enddo
c solid eqns. for a typical node
p=0.d0
  j=nnf+ncz+2
m=j+ncz-1
do i=1,ncr
   do i1=j,m
  n=i1+1
  s=i1-1
  w=i1-ncz-2
  e=i1+ncz+2
  j1=i1-nnf
  drw=dr
  dre=dr
  dzn=dz
  dzs=dz
  if(i .eq. 1) then
   w=i1-ncz-1
   drw=dr/2.d0
  elseif(i .eq. ncr) then
   e=i1+ncz+1
   dre=dr/2.d0
  endif
  if(i1 .eq. j) then
   dzs=dz/2.d0
  elseif(i1 .eq. m) then
   dzn=dz/2.d0
```

```
endif
tp=temp(i1)
te=temp(e)
   tw=temp(w)
tn=temp(n)
ts=temp(s)
tm=temp(j1)
cps=cpspt(tp,pf)
ce=(cons(tp,pf)+cons(te,pf))/2.d0
cw=(cons(tp,pf)+cons(tw,pf))/2.d0
cn=(cons(tp,pf)+cons(tn,pf))/2.d0
cs=(cons(tp,pf)+cons(ts,pf))/2.d0
ta=tavesf(i)
rhog=rhof(ta)
mug=viscf(ta)
  re=dp*rhog*rhs(nnt+i+1)/mug
      pr=cpgas(tm)*mug/cgas(tm)
      nu=2.d0+1.2d0*re**(0.5d0)*pr**(1.d0/3.d0)
      h1=nu*cgas(tm)/dp
r=(p+5.d-1)*dr
      aa=6.d0*(1.d0-eps(r))/dp
      h1=h1*aa
qgen=dabs(2.d0*pi*frequ*efs(j1)**2*eppm*eps0)
r=(p+1.d0)*dr
a(i1,e)=-(1.d0-eps(r))*ce*(p+1.d0)*dr*dz/dre
r=p*dr
a(i1,w)=-(1.d0-eps(r))*cw*p*dr*dz/drw
r=(p+5.d-1)*dr
a(i1,n)=-(1.d0-eps(r))*cn*((p+1.d0)**2-p**2)/2.d0*dr**2/dzn
a(i1,s)=-(1.d0-eps(r))*cs*((p+1.d0)**2-p**2)/2.d0*dr**2/dzs
a(i1,j1)=-h1*((p+1.d0)**2-p**2)/2.d0*dr**2*dz
a(i1,i1)=rhos*cps*(1.d0-eps(r))*((p+1.d0)**2-p**2)/2.d0*dr**2*
```

```
1           dz/dt-a(i1,e)-a(i1,w)-a(i1,n)-a(i1,s)-a(i1,j1)
  rhs(i1)=(1.d0-eps(r))*((p+1.d0)**2-p**2)/2.d0*dr**2*dz*
1          (rhos*cps/dt*tprev(i1)+qgen)
 enddo
 j=j+ncz+2
 m=m+ncz+2
 p=p+1.d0
enddo
c fluid equations
c
c left boundary
      a(nnt+1,nnt+1)=1.d0
a(nnt+1,nnt+2)=-1.d0
rhs(nnt+1)=0.d0
c right boundary
      a(nnt+ncr+2,nnt+ncr+2)=1.d0
rhs(nnt+ncr+2)=0.d0
c typical node
      p=0.d0
      do i=nnt+2,nnt+ncr+1
 e=i+1
 w=i-1
 dre=dr
 drw=dr
 if(i .eq. nnt+2) then
  drw=dr/2.d0
 elseif(i .eq. nnt+ncr+1) then
  dre=dr/2.d0
 endif
 tp=tavesf(i-nnt-1)
 rhog=rhof(tp)
 mug=viscf(tp)
```

```
 r=(p+1.d0)*dr
 a(i,i+1)=-r*eps(r)*mug/dre
 r=p*dr
 a(i,i-1)=-r*eps(r)*mug/drw
 r=(p+5.d-1)*dr
 vol=((p+1.d0)**2-p**2)/2.d0*dr**2
 f1=1.5d2*mug*(1.d0-eps(r))**2/(dp**2*eps(r)**3)
 f2=1.75d0*rhog*(1.d0-eps(r))/(dp*eps(r)**3)
 a(i,i)=-a(i,i+1)-a(i,i-1)+(f1+f2*vguess(i-nnt))*vol
 rhs(i)=-dpdz*vol
 p=p+1.d0
enddo
c
return
end
c
function eps(r)
implicit double precision(a-h,o-y)
real*8 kg,n,mu0
integer flag,flag2,flag3
c
common/param/flag,flag2,flag3
common/dielectric/pi,frequ,eps0,c0,mu0,ef0
common/solgas/rhos,dp,pf,dpt,vf,tmpin,kg,epsi,rb,ro,hn
common/delta/h,dt,dr,dz
c
      c=1.d0
n=2.d0
eps=epsi*(1.d0+c*dexp(-n*(rb-r)/dp))
c eps=epsi
```

Appendix B

Fluidized Bed Program

```
c    This program will determine the temperature and velocity profiles and
c    the temperature difference between the solid and gas in a 2-d fluidized
c    bed.
c
c    The following is a definition of the input variables.
c
c     initer = number of inner iterations.
c     ncx = number of cells in the x-direction.
c     ncy = number of cells in the flow, or y-direction in the expanded bed
c           region.
c     ncyp = total number of cells in the y-direction.
c     nnf = total number of nodes for a single phase.
c     nnt = total number of nodes for both phases.
c     ntime = number of time steps.
c     job = parameter set to zero to solve Ax = b in the matrix solver.
c     kg = conductivity of glass (W/m\cdot K).
c     hn = heat transfer coefficient between container and surroundings
c          (W/$m^2$\cdot K).
c     len = length of the bed (m).
c     lout = thickness of the bed container (m).
```

```
c     h = height of the bed (m).
c     dx = length of a control volume in the computational grid (m).
c     dy = height of a control volume in the computational grid (m).
c     dt = time step (s).
c     dpt = particle diameter of the metallic catalyst particle (m).
c     pf = volume fraction of metallic catalyst particle in the ceramic
c          support.
c     rhoal = density of alumina (kg/$m^3$).
c     rhopt = density of platinum (kg/$m^3$).
c     vin = inlet gas velocity (m/s).
c     vflow = inlet volume flow rate of the gas ($m^3$/s).
c     dp = particle diameter (m).
c     rhos = density of the ceramic support/metallic catalyst mixture
c            (kg/$m^3$).
c     tin = inlet gas temperature (K).
c     frequ = microwave frequency (Hz).
c     eps0 = permitivity of free space (f/m).
c     mu0 = permeability of free space (h/m).
c     c0 = speed of light (m/s).
c     ef0 = electric field within the bed (V/m).
c     rhog = density of the gas (kg/$m^3$).
c     mug = dynamic viscosity of the gas (N\cdot s/$m^2$).
c     eps = solid volume fraction array.
c     vxa = gas x-velocity array.
c     vxs = solid x-velocity array.
c     vya = gas y-velocity array.
c     vys = solid y-velocity array.
c
c     flag = determines whether the support will be alpha or gamma-alumina:
c            if set to 1, then alpha alumina is used; otherwise gamma
c            alumina is used.
c
```

```
c     flag1 = controls the numerical scheme: if set to 1, the central
c             difference scheme is implemented; if set to 2, the upwind
c             scheme is imlemented; if set to 3, the hybrid scheme is
c             implemented; and if set to 4, the power law is implemented.
c
c     flag2 = determines the inlet boundary condition for the solid phase:
c             if set to 1, the solid is insulated; otherwise the solid is
c             set to the inlet temperature of the gas.
c
c     flag3 = controls electric field: if set to 1, the electric field
c             varies linearly in the flow direction; otherwise, the electric
c             field is uniform.
c
      implicit real*8(a-h,o-y)
      real*8 k1,kg,mug,mu0,z,len,lout
      integer ncx,ncy,nnf,nnt,ntime,flag,flag1,flag2,flag3,nn,lda,ipvt
integer job,nill,initer,flag4,flag5
c
      parameter(initer=4)
parameter(ncx=50)
      parameter(ncy=21)
parameter(nill=(ncx+1)*(ncy+1))
      parameter(nnf=2*(ncy+ncx)+ncx*ncy)
      parameter(nnt=2*nnf)
      parameter(nn=nnt)
      parameter(lda=nn)
c
      dimension tprev(nnt),z(nn),ipvt(nn),temp(nnt),a(lda,nn),rhs(nnt)
dimension taves(ncy),efelds(ncy),efs(nnf),vya(nill),vxa(nill)
dimension vys(nill),vxs(nill),eps(nill),epf(nill)
c
common/param/flag,flag1,flag2,flag4,flag5
```

```
common/dielectric/pi,freq,eps0,c0,mu0,ef0
common/solgas/rhog,mug,vin,rhos,dp,pf,dpt,vf,tin,qflux
common/delta/dt,dx,dy,h,lout,hn,kg
c
open(1,file='c:\fbed\eps_100_.033')
open(2,file='c:\fbed\yair_100_.033')
open(3,file='c:\fbed\xair_100_.033')
open(4,file='c:\fbed\ysol_100_.033')
open(5,file='c:\fbed\xsol_100_.033')
c
c      open(1,file='c:\fbed\eps_125_.051')
c open(2,file='c:\fbed\yair_125_.051')
c open(3,file='c:\fbed\xair_125_.051')
c open(4,file='c:\fbed\ysol_125_.051')
c open(5,file='c:\fbed\xsol_125_.051')
c
c      open(1,file='c:\fbed\eps_150_.049')
c open(2,file='c:\fbed\yair_150_.049')
c open(3,file='c:\fbed\xair_150_.049')
c open(4,file='c:\fbed\ysol_150_.049')
c open(5,file='c:\fbed\xsol_150_.049')
c
c      open(1,file='c:\fbed\eps_150_.073')
c open(2,file='c:\fbed\yair_150_.073')
c open(3,file='c:\fbed\xair_150_.073')
c open(4,file='c:\fbed\ysol_150_.073')
c      open(5,file='c:\fbed\xsol_150_.073')
c
open(6,file='c:\fbed\length')
      open(7,file='c:\fbed\height')
      open(8,file='c:\fbed\time')
c
```

```
      open(9,file='c:\fbed\tfx_100_33')
      open(10,file='c:\fbed\tsx_100_33')
      open(11,file='c:\fbed\tfy_100_33')
      open(12,file='c:\fbed\tsy_100_33')
open(14,file='c:\fbed\xvela_100_33')
open(15,file='c:\fbed\xvels_100_33')
open(16,file='c:\fbed\yvela_100_33')
open(17,file='c:\fbed\yvels_100_33')
open(18,file='c:\fbed\epsilon_100_33')
c
c      open(9,file='c:\fbed\tfx_125_51')
c      open(10,file='c:\fbed\tsx_125_51')
c      open(11,file='c:\fbed\tfy_125_51')
c      open(12,file='c:\fbed\tsy_125_51')
c open(14,file='c:\fbed\xvela_125_51')
c open(15,file='c:\fbed\xvels_125_51')
c open(16,file='c:\fbed\yvela_125_51')
c open(17,file='c:\fbed\yvels_125_51')
c open(18,file='c:\fbed\epsilon_125_51')
c
c      open(9,file='c:\fbed\tfx_150_495')
c open(10,file='c:\fbed\tsx_150_495')
c open(11,file='c:\fbed\tfy_150_495')
c open(12,file='c:\fbed\tsy_150_495')
c open(14,file='c:\fbed\xvela_150_49')
c open(15,file='c:\fbed\xvels_150_49')
c open(16,file='c:\fbed\yvela_150_49')
c open(17,file='c:\fbed\yvels_150_49')
c open(18,file='c:\fbed\epsilon_150_49')
c
c      open(9,file='c:\fbed\tfx_150_73')
c open(10,file='c:\fbed\tsx_150_73')
```

```
c open(11,file='c:\fbed\tfy_150_73')
c open(12,file='c:\fbed\tsy_150_73')
c      open(14,file='c:\fbed\xvela_150_73')
c      open(15,file='c:\fbed\xvels_150_73')
c      open(16,file='c:\fbed\yvela_150_73')
c      open(17,file='c:\fbed\yvels_150_73')
c open(18,file='c:\fbed\epsilon_150_73')
c
open(19, file='c:\fbed\tgas_100_33')
open(20, file='c:\fbed\tsol_100_33')
open(21, file='c:\fbed\tdiff_100_33')
c
      ntime=600
      pi=dacos(-1.d0)
      job=0
hn=2.d0
kg=1.4d0
k1=1.d5
qflux=0.d0
ncyp=50
vin=3.3d-2
      dp=1.d-4
len=4.d-1
lout=4.d-2
h=6.d-1
dx=len/dble(ncx)
dy=h/dble(ncyp)
      dt=1.d1
      dpt=4.d-8
      pf=.01d0
      rhoal=7.d2
      rhopt=21450.d0
```

```
      rhog=1.2d0
mug=1.8e-5
      vf=(pf/rhopt)/((pf/rhopt)+(1.d0-pf)/rhoal)
rhos=pf*rhopt+(1.d0-pf)*rhoal
      tin=300.15d0
      freq=.915d9
      eps0=8.854d-12
      mu0=4.d-7*pi
c0=3.d8
ef0=3.d3
flag=2
flag1=2
flag2=1
flag3=2
flag4=2
flag5=2
c efelds(ncy)=ef0
tol=1.d-4
c
      do i=1,nnt
       tprev(i)=tin
 temp(i)=tin
      enddo
c
do i=1,nill
 read(1,*) eps(i)
 epf(i)=1.d0-eps(i)
 read(2,*) vya(i)
 read(3,*) vxa(i)
 read(4,*) vys(i)
 read(5,*) vxs(i)
enddo
```

```
c
      do i2=1,ntime
       k=nnf+ncx+2
       l=k+ncx-1
       do i=1,ncy
  sum=0.d0
        do j=k,l
   sum=sum+temp(j)
  enddo
  taves(i)=sum/dble(ncx)
  k=k+ncx+2
  l=l+ncx+2
 enddo
c
       call efields(k1,taves,efelds,eppm,ncy)
c
       do k=1,nnt
        temp(k)=tprev(k)
       enddo
c
       k=ncx+2
       l=k+ncx-1
       do i=1,ncy
        do j=k,l
   if(flag3 .eq. 1) then
    efs(j)=efelds(i)
   else
    efs(j)=ef0
   endif
  enddo
  k=k+ncx+2
  l=l+ncx+2
```

```
 enddo
c
    5  do j2=1,initer
c
        call assemble(tprev,a,rhs,vya,vxa,vys,vxs,eps,temp,efs,eppm,
1               lda,nn,ncx,ncy,nnf,nnt,nill,epf)
c   if(i2.eq.ntime) then
c     do j3=1,lda
c     do j4=1,nn
c      write(*,*) j3, j4, a(j3,j4), rhs(j3)
c     enddo
c    enddo
c    do j3=1,nnt
c         write(*,*) i2, j3, rhs(j3)
c    enddo
c   endif
  call dgeco(a,lda,nn,ipvt,rcond,z)
c   write(*,*) 'the condition # is', 1.d0/rcond, i2
c   if(1.d0/rcond .gt. 1.d3) then
c    write(*,*) 1.d0/rcond, 'is too large at the time-step', i2
c    goto 30
c   endif
  call dgesl(a,lda,nn,ipvt,rhs,job)
c   if(i2.gt.2000) then
c          do j3=1,nnt
c        write(6,*) i2, j3, rhs(j3)
c     enddo
c   endif
c
        do l=1,nnt
         temp(l)=rhs(l)
        enddo
```

```
  do i3=1,lda
   do j3=1,nn
    a(i3,j3)=0.d0
   enddo
  enddo
      enddo
c
c       if(mod(i2,tint) .eq. 1) then
c   write(8,18) dble(i2)*dt
        write(9,11) (rhs(l),l=int(ncy/2)*(ncx+2)-1,int(ncy/2)*(ncx+2)+
1              ncx)
        write(10,11) (rhs(l),l=nnf+int(ncy/2)*(ncx+2)-1,nnf+int(ncy/2)*
1              (ncx+2)+ncx)
        write(11,15) rhs(int(ncx/2)),(rhs(l),l=ncx+1+int(ncx/2),ncx+1+
1              int(ncx/2)+(ncy-1)*(ncx+2),ncx+2),rhs(2*(ncx+1)+
1              int(ncx/2)+(ncy-1)*(ncx+2))
        write(12,15) rhs(nnf+int(ncx/2)),(rhs(l),l=nnf+ncx+1+
1              int(ncx/2),nnf+ncx+1+int(ncx/2)+(ncy-1)*(ncx+2),
     1              ncx+2),rhs(nnf+2*(ncx+1)+int(ncx/2)+(ncy-1)*(ncx+2))
c
   11   format(1x,52(f7.2,1x))
   15   format(1x,23(f7.2,3x))
c       endif
       if(i2 .eq. 1) then
  do l=1,ncx+2
   if(l .eq. 1) then
    write(6,17) 0.d0
   elseif(l .eq. ncx+2) then
    write(6,17) 1.d0
   else
         write(6,17) (dble(l)-1.5d0)*dx/len
   endif
```

```
  enddo
  h=6.d-1
        do l=1,ncy+2
   if(l .eq. 1) then
    write(7,17) 0.d0
   elseif(l .eq. ncy+2) then
    write(7,17) 1.d0
   else
          write(7,17) (dble(l)-1.5d0)*dy/h
   endif
        enddo
   17   format(f6.4)
  do l=1,ntime
   write(8,18) dble(l)*dt
  enddo
   18   format(f6.1)
       endif
c
       if(i2 .lt. ntime) then
        do l=1,nnt
         tprev(l)=rhs(l)
        enddo
       endif
      enddo
c
k=1
l=ncx+1
do i=1,ncy+1
 write(14,19) (vxa(m), m=k,l)
 write(15,19) (vxs(m), m=k,l)
 write(16,20) (vya(m), m=k,l)
 write(17,20) (vys(m), m=k,l)
```

```
 write(18,21) (eps(m), m=k,l)
 k=k+ncx+1
 l=l+ncx+1
enddo
   19 format(1x,51(f9.6,3x))
   20 format(1x,51(f9.6,3x))
   21 format(1x,51(f8.6,3x))
c
      k=0
      do l=1,ncy+2
 if(l .eq. 1) then
  write(19,23) (rhs(m), m=1,ncx)
  write(20,23) (rhs(m+nnf), m=1,ncx)
  write(21,25) (rhs(m+nnf)-rhs(m), m=1,ncx)
 elseif(l .eq. ncy+2) then
  write(19,23) (rhs(m), m=nnf-ncx+1,nnf)
  write(20,23) (rhs(m+nnf), m=nnf-ncx+1,nnf)
  write(21,25) (rhs(m+nnf)-rhs(m), m=nnf-ncx+1,nnf)
 else
  write(19,22) (rhs(m),m=ncx+1+k,2*(ncx+1)+k)
  write(20,22) (rhs(m+nnf),m=ncx+1+k,2*(ncx+1)+k)
  write(21,24) (rhs(m+nnf)-rhs(m), m=ncx+1+k,2*(ncx+1)+k)
 endif
 if(l .gt. 1) then
  k=k+ncx+2
 endif
enddo
c
   22 format(1x,52(f7.2,3x))
   23 format(1x,50(f7.2,3x))
   24 format(1x,52(f9.4,3x))
   25 format(1x,50(f9.4,3x))
```

```
c
      close(1)
      close(2)
      close(3)
      close(4)
      close(5)
      close(6)
      close(7)
      close(8)
      close(9)
      close(10)
      close(11)
      close(12)
close(14)
close(15)
close(16)
close(17)
close(18)
close(19)
close(20)
close(21)
   30 end
c
      subroutine assemble(tprev,a,rhs,vya,vxa,vys,vxs,eps,temp,efs,
1                   eppm,lda,nn,ncx,ncy,nnf,nnt,nill,epf)
      implicit real*8(a-h,o-y)
      real*8 kg,mug,mu0,nu,lout
      integer flag,flag1,flag2,flag4,flag5,e,s,w,n,nn,lda,ncx,ncy
integer nnf,nnt,nill
c
      dimension a(lda,nn),temp(nnt),tprev(nnt),rhs(nnt),eps(nill)
      dimension vya(nill),vxa(nill),vys(nill),vxs(nill),efs(nnf)
```

```
dimension epf(nill)
c
common/param/flag,flag1,flag2,flag4,flag5
common/dielectric/pi,freq,eps0,c0,mu0,ef0
common/solgas/rhog,mug,vin,rhos,dp,pf,dpt,vf,tin,qflux
common/delta/dt,dx,dy,h,lout,hn,kg
c gas boundary nodes
k=1
do i=ncx+1,nnf-2*ncx-1,ncx+2
 eps1=(epf(i+k)+epf(i-ncx-1+k))/2.d0
 tp=temp(i)
 rtot=lout/kg+1.d0/hn
 a(i,i)=1.d0/rtot+eps1*cgas(tp)/dx
 a(i,i+1)=-eps1*cgas(tp)/dx
 rhs(i)=tin/rtot
 k=k+1
enddo
c
      k=0
do i=2*(ncx+1),nnf-ncx,ncx+2
 eps1=(epf(i+k)+epf(i-ncx-1+k))/2.d0
 tp=temp(i)
 rtot=lout/kg+1.d0/hn
 a(i,i)=1.d0/rtot+eps1*cgas(tp)/dx
 a(i,i-1)=-eps1*cgas(tp)/dx
 rhs(i)=tin/rtot
 k=k+1
      enddo
c
do i=nnf-ncx+1,nnf
 a(i,i)=1.d0
 a(i,i-ncx-1)=-1.d0
```

```
 rhs(i)=0.d0
enddo
c gas inlet temperature boundary
      do i=1,ncx
 a(i,i)=1.d0
 rhs(i)=tin
enddo
c gas eqns. for a typical node
      nt=0
nt1=ncx+1
j=ncx+2
m=2*ncx+1
do i=1,ncy
   do i1=j,m
  n=i1+ncx+2
  s=i1-ncx-2
  w=i1-1
  e=i1+1
  j1=i1+nnf
  fee=.5d0
  few=.5d0
  fen=.5d0
  fes=.5d0
  ee=(epf(i1-nt1+1)+epf(i1-nt+1))/2.d0
  ew=(epf(i1-nt1)+epf(i1-nt))/2.d0
  en=(epf(i1-nt)+epf(i1-nt+1))/2.d0
  es=(epf(i1-nt1)+epf(i1-nt1+1))/2.d0
  epfi1=(epf(i1-nt1+1)+epf(i1-nt+1)+epf(i1-nt1)+epf(i1-nt))/4.d0
  epfe=(epf(i1+1-nt1+1)+epf(i1+1-nt+1)+epf(i1+1-nt1)+
1        epf(i1+1-nt))/4.d0
  epfw=(epf(i1-1-nt1+1)+epf(i1-1-nt+1)+epf(i1-1-nt1)+
1        epf(i1-1-nt))/4.d0
```

```
  epfn=(epf(i1+ncx+2-(nt1+1)+1)+epf(i1+ncx+2-(nt+1)+1)+
1        epf(i1+ncx+2-(nt1+1))+epf(i1+ncx+2-(nt+1)))/4.d0
  epfs=(epf(i1-ncx-2-(nt1-1)+1)+epf(i1-ncx-2-(nt-1)+1)+
1        epf(i1-ncx-2-(nt1-1))+epf(i1-ncx-2-(nt-1)))/4.d0
  if(i .eq. 1) then
   s=i1-ncx-1
   fes=0.d0
   epfs=es
  elseif(i .eq. ncy) then
   n=i1+ncx+1
   fen=0.d0
   epfn=en
  endif
  if(i1 .eq. j) then
   dfw=0.d0
   epfw=ew
  elseif(i1 .eq. m) then
   dfe=0.d0
   epfe=ee
  endif
  tp=temp(i1)
    te=temp(e)
    tw=temp(w)
  tn=temp(n)
  ts=temp(s)
  tm=temp(j1)
  cpg=cpgas(tp)
  ue=(vxa(i1-nt1+1)+vxa(i1-nt+1))/2.d0
  uw=(vxa(i1-nt1)+vxa(i1-nt))/2.d0
  vn=(vya(i1-nt)+vya(i1-nt+1))/2.d0
  vs=(vya(i1-nt1)+vya(i1-nt1+1))/2.d0
  cee=1.d0/((1.d0-fee)/(epfi1*cgas(tp))+fee/(epfe*cgas(te)))
```

```
  cwe=1.d0/((1.d0-few)/(epfi1*cgas(tp))+few/(epfw*cgas(tw)))
  cne=1.d0/((1.d0-fen)/(epfi1*cgas(tp))+fen/(epfn*cgas(tn)))
  cse=1.d0/((1.d0-fes)/(epfi1*cgas(tp))+fes/(epfs*cgas(ts)))
  fe=rhog*ee*ue*dy
  fw=rhog*ew*uw*dy
  fn=rhog*en*vn*dx
  fs=rhog*es*vs*dx
  de=cee/cpg*dy/dx
  dw=cwe/cpg*dy/dx
  dn=cne/cpg*dx/dy
  ds=cse/cpg*dx/dy
  pe=fe/de
  pw=fw/dw
  pn=fn/dn
  ps=fs/ds
  vvi1=(vya(i1-nt1+1)+vya(i1-nt+1)+vya(i1-nt1)+vya(i1-nt))/4.d0
  vvj1=(vys(i1-nt1+1)+vys(i1-nt+1)+vys(i1-nt1)+vys(i1-nt))/4.d0
  vui1=(vxa(i1-nt1+1)+vxa(i1-nt+1)+vxa(i1-nt1)+vxa(i1-nt))/4.d0
  vuj1=(vxs(i1-nt1+1)+vxs(i1-nt+1)+vxs(i1-nt1)+vxs(i1-nt))/4.d0
  vmag=dsqrt((vvi1-vvj1)**2+(vui1-vuj1)**2)
c   re=(1.d0-epfi1)*dp*rhog*vmag/mug
  re=dp*rhog*vin/mug
  nu=3.d-2*re**(1.3d0)
c   pr=cpg*mug/cgas(tp)
c        nu=2.d0+1.2d0*re**(0.5d0)*pr**(1.d0/3.d0)
        h1=nu*cgas(tp)/dp
        aa=6.d0*(1.d0-epfi1)/dp
        h1=h1*aa
  a(i1,e)=-(de*ap(pe,flag1)+dmax1(-fe,0.d0))
  a(i1,w)=-(dw*ap(pw,flag1)+dmax1(fw,0.d0))
  a(i1,n)=-(dn*ap(pn,flag1)+dmax1(-fn,0.d0))
  a(i1,s)=-(ds*ap(ps,flag1)+dmax1(fs,0.d0))
```

```
  a(i1,j1)=-h1*dx*dy/cpg
  a(i1,i1)=rhog*epfi1*dx*dy/dt-a(i1,e)-a(i1,w)-a(i1,n)-
1           a(i1,s)-a(i1,j1)
  rhs(i1)=epfi1*rhog*dx*dy/dt*tprev(i1)
 enddo
 j=j+ncx+2
 m=m+ncx+2
 nt=nt+1
 nt1=nt1+1
enddo
c solid boundary nodes
k=1
do i=nnf+ncx+1,2*nnf-2*ncx-1,ncx+2
 eps1=1.d0-(epf(i-nnf+k)+epf(i-nnf-ncx-1+k))/2.d0
 tp=temp(i)
 rtot=lout/kg+1.d0/hn
 a(i,i)=1.d0/rtot+eps1*cons(tp,pf)/dx
 a(i,i+1)=-eps1*cons(tp,pf)/dx
 rhs(i)=tin/rtot
 k=k+1
enddo
c
      k=0
do i=nnf+2*(ncx+1),2*nnf-ncx,ncx+2
 eps1=1.d0-(epf(i-nnf+k)+epf(i-nnf-ncx-1+k))/2.d0
 tp=temp(i)
 rtot=lout/kg+1.d0/hn
 a(i,i)=1.d0/rtot+eps1*cons(tp,pf)/dx
 a(i,i-1)=-eps1*cons(tp,pf)/dx
 rhs(i)=tin/rtot
 k=k+1
      enddo
```

```
c
nt=1
      do i=2*nnf-ncx+1,2*nnf
c    if(flag4 .eq. 2) then
    a(i,i)=1.d0
    a(i,i-ncx-1)=-1.d0
    rhs(i)=0.d0
c    else
c    tp=temp(i)
c   tm=temp(i-nnf)
c   h2=2.d0*cgas(tm)/dp
c   epsi1=(eps(ncy*(ncx+1)+nt+1)+eps(ncy*(ncx+1)+nt))/2.d0
c   a(i,i)=h2*dx+epsi1*cons(tp,pf)
c   a(i,i-ncx-1)=-epsi1*cons(tp,pf)
c   rhs(i)=h2*dx*tm
c   nt=nt+1
c    endif
  enddo
c
  if(flag2 .eq. 1) then
c heat flux boundary
       do i=nnf+1,nnf+ncx
c     if(flag5 .eq. 2) then
     a(i,i)=1.d0
     a(i,i+ncx+1)=-1.d0
           rhs(i)=0.d0
c     else
c      tp=temp(i)
c      tm=tin
c      h2=2.d0*cgas(tm)/dp
c      epsi1=(eps(i-nnf)+eps(i+1-nnf))/2.d0
c      a(i,i)=h2*dx+epsi1*cons(tp,pf)
```

```
c      a(i,i+ncx+1)=-epsi1*cons(tp,pf)
c      rhs(i)=h2*dx*tm
c     endif
   enddo
  else
c temperature boundary
  do i=nnf+1,nnf+ncx
        a(i,i)=1.d0
   rhs(i)=tin
   enddo
  endif
c solid eqns. for a typical node
       nt=0
  nt1=ncx+1
    j=nnf+ncx+2
        m=j+ncx-1
  do i=1,ncy
    do i1=j,m
  n=i1+ncx+2
  s=i1-ncx-2
  w=i1-1
  e=i1+1
  j1=i1-nnf
  fee=.5d0
  few=.5d0
  fen=.5d0
  fes=.5d0
  ee=(eps(j1-nt1+1)+eps(j1-nt+1))/2.d0
  ew=(eps(j1-nt1)+eps(j1-nt))/2.d0
  en=(eps(j1-nt)+eps(j1-nt+1))/2.d0
  es=(eps(j1-nt1)+eps(j1-nt1+1))/2.d0
  epsi1=(eps(j1-nt1+1)+eps(j1-nt+1)+eps(j1-nt1)+eps(j1-nt))/4.d0
```

```
  epse=(eps(j1+1-nt1+1)+eps(j1+1-nt+1)+eps(j1+1-nt1)+
1        eps(j1+1-nt))/4.d0
  epsw=(eps(j1-1-nt1+1)+eps(j1-1-nt+1)+eps(j1-1-nt1)+
1        eps(j1-1-nt))/4.d0
  epsn=(eps(j1+ncx+2-(nt1+1)+1)+eps(j1+ncx+2-(nt+1)+1)+
1        eps(j1+ncx+2-(nt1+1))+eps(j1+ncx+2-(nt+1)))/4.d0
  epss=(eps(j1-ncx-2-(nt1-1)+1)+eps(j1-ncx-2-(nt-1)+1)+
1        eps(j1-ncx-2-(nt1-1))+eps(j1-ncx-2-(nt-1)))/4.d0
  if(i .eq. 1) then
   s=i1-ncx-1
   fes=0.d0
   epss=es
  elseif(i .eq. ncy) then
   n=i1+ncx+1
   fen=0.d0
   epsn=en
  endif
  if(i1 .eq. j) then
   few=0.d0
   epsw=ew
  elseif(i1 .eq. m) then
   fee=0.d0
   epse=ee
  endif
  tp=temp(i1)
  te=temp(e)
     tw=temp(w)
  tn=temp(n)
  ts=temp(s)
  tm=temp(j1)
  cps=cpspt(tp,pf)
  ue=(vxs(j1-nt1+1)+vxs(j1-nt+1))/2.d0
```

```
  uw=(vxs(j1-nt1)+vxs(j1-nt))/2.d0
  vn=(vys(j1-nt)+vys(j1-nt+1))/2.d0
  vs=(vys(j1-nt1)+vys(j1-nt1+1))/2.d0
  cee=1.d0/((1.d0-fee)/(epsi1*cons(tp,pf))+fee/(epse*cons(te,pf)))
  cwe=1.d0/((1.d0-few)/(epsi1*cons(tp,pf))+few/(epsw*cons(tw,pf)))
  cne=1.d0/((1.d0-fen)/(epsi1*cons(tp,pf))+fen/(epsn*cons(tn,pf)))
  cse=1.d0/((1.d0-fes)/(epsi1*cons(tp,pf))+fes/(epss*cons(ts,pf)))
  fe=rhos*ee*ue*dy
  fw=rhos*ew*uw*dy
  fn=rhos*en*vn*dx
  fs=rhos*es*vs*dx
  de=cee/cps*dy/dx
  dw=cwe/cps*dy/dx
  dn=cne/cps*dx/dy
  ds=cse/cps*dx/dy
  pe=fe/de
  pw=fw/dw
  pn=fn/dn
  ps=fs/ds
  vvi1=(vya(j1-nt1+1)+vya(j1-nt+1)+vya(j1-nt1)+vya(j1-nt))/4.d0
  vvj1=(vys(j1-nt1+1)+vys(j1-nt+1)+vys(j1-nt1)+vys(j1-nt))/4.d0
  vui1=(vxa(j1-nt1+1)+vxa(j1-nt+1)+vxa(j1-nt1)+vxa(j1-nt))/4.d0
  vuj1=(vxs(j1-nt1+1)+vxs(j1-nt+1)+vxs(j1-nt1)+vxs(j1-nt))/4.d0
  vmag=dsqrt((vvi1-vvj1)**2+(vui1-vuj1)**2)
c   re=epsi1*dp*rhog*vmag/mug
  re=dp*rhog*vin/mug
  nu=3.d-2*re**(1.3d0)
c   pr=cpg*mug/cgas(tp)
c        nu=2.d0+1.2d0*re**(0.5d0)*pr**(1.d0/3.d0)
        h1=nu*cgas(tm)/dp
        aa=6.d0*epsi1/dp
        h1=h1*aa
```

```
  qgen=dabs(2.d0*pi*freq*efs(j1)**2*eppm*eps0)
  a(i1,e)=-(de*ap(pe,flag1)+dmax1(-fe,0.d0))
  a(i1,w)=-(dw*ap(pw,flag1)+dmax1(fw,0.d0))
  a(i1,n)=-(dn*ap(pn,flag1)+dmax1(-fn,0.d0))
  a(i1,s)=-(ds*ap(ps,flag1)+dmax1(fs,0.d0))
  a(i1,j1)=-h1*dx*dy/cps
  a(i1,i1)=rhos*epsi1*dx*dy/dt-a(i1,e)-a(i1,w)-a(i1,n)-a(i1,s)-
1           a(i1,j1)
  rhs(i1)=epsi1*rhos*dx*dy/dt*tprev(i1)+epsi1*qgen*dx*dy/cps
 enddo
 j=j+ncx+2
 m=m+ncx+2
 nt=nt+1
 nt1=nt1+1
enddo
return
end
c
      function ap(p,flag1)
      implicit real*8(a-h,o-y)
integer flag1
c
if(flag1 .eq. 1) then
c  central difference
 ap=1.d0-dabs(p)/2.d0
elseif(flag1 .eq. 2) then
c  upwind
       ap=1.d0
elseif(flag1 .eq. 3) then
c  hybrid
       ap=dmax1(0.d0,1.d0-5d-1*dabs(p))
elseif(flag1 .eq. 4) then
```

```
c  power law
       ap=dmax1(0.d0, (1.d0-1d-1*dabs(p))**5)
endif
c
return
end
```

Appendix C

Subroutines and Functions Used By the Packed and Fluidized Bed Programs

```
c
       function cap1(t1)
c
c     ...heat capacity for alumina
c
      implicit double precision(a-h,o-y)
      dimension a(6)
      data a/-210.672d0,5.1956d0,-7.95646d-03,6.23173d-06,
     &  -2.40174d-09,3.61348d-13/
        sum=0.0d0
        do 5 i=1,5
        sum=t1*(sum+a(7-i))
    5   continue
        cap1=sum+a(1)
      return
      end
c
```

```
      function cpt(t)
c
c          29 June 1995
c
c     ... heat capcity for platinum
c
      implicit real*8(a-h,o-y)
      real*8 cpt,cpts
      dimension tpts(9),cpts(9)
      data tpts/2.d+02,3.d+02,4.d+02,6.d+02,8.d+02,1.d+03,1.2d+03
     2,1.5d+03,2.d+03/
      data cpts/1.25d2,1.33d2,1.36d2,1.41d2,1.46d2,1.52d2,
     2 1.57d2,1.65d2,1.79d2/
      if(t.ge.tpts(9))then
          a=(cpts(9)-cpts(8))/(tpts(9)-tpts(8))
          b=cpts(9)-a*tpts(9)
          cpt=a*t+b
          return
      endif
      do 5 i=1,8
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          cpt=cpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          cpt=cpts(i)+(t-x)*(cpts(i+1)-cpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
```

```
c
       function cal(x1)
c
c     ...conductivity for alumina
c
c
      implicit double precision(a-h,o-y)
      cal=0.55d0*(6.788d+03/(x1-125.0d0)+(3.562d-33)*x1**10)
      return
      end
c
      function cpspt(t,pf)
c
c          29 June 95
c
c     ... heat capacity for catalyst pellets
c
      implicit real*8(a-h,o-y)
      cpspt=(1.d0-pf)*cap1(t)+pf*cpt(t)
      return
      end
c
      subroutine efields(k1,taves,efelds,eppm,ncy)
      implicit real*8(a-h,o-y)
      implicit complex*16(z)
      real*8 kg,k1,ks1,ks2,ks3,ks4,mu0,mug,lout
      integer flag,ncy,flag1,flag2,flag4,flag5
c
      dimension efelds(ncy),taves(ncy)
c
common/param/flag,flag1,flag2,flag4,flag5
common/dielectric/pi,freq,eps0,c0,mu0,ef0
```

```
common/solgas/rhog,mug,vin,rhos,dp,pf,dpt,vf,tin,qflux
common/delta/dt,dx,dy,h,lout,hn,kg
c
c      do k=ncy,1,-1
do k=1,ncy
       tp=taves(k+1)
   ef=efelds(k+1)
       if(flag .eq. 1) then
        zep1=dcmplx(epa(tp),-eppa(tp))
       else
        zep1=dcmplx(epag(tp),-eppag(tp))
       endif
c
       zep2=dcmplx(1.d0,-eppt(tp,dpt))
       zepm=((2.d0*vf+1.d0)*zep1*zep2+2.d0*zep1**2*(1.d0-vf))/
     1      ((2.d0+vf)*zep1+(1.d0-vf)*zep2)
       eppm=dimag(zepm)
 epm=dreal(zepm)
 tandm=eppm/epm
 efelds(k)=ef0+k1*(dble(k)-.5d0)*dy
c  beta=freq*dsqrt(epm)/c0*dsqrt(.5d0*(dsqrt(1.d0+tandm**2)-1.d0))
c  ks1=fes(beta,ef)
c       ks2=fes(beta,ef+.5d0*dy*ks1)
c  ks3=fes(beta,ef+.5d0*dy*ks2)
c  ks4=fes(beta,ef+dy*ks3)
c  efelds(k)=efelds(k+1)+dy/6.d0*(ks1+2.d0*ks2+2.d0*ks3+ks4)
enddo
c
      return
      end
c
      function fes(beta,ef)
```

```
implicit real*8(a-h,o-y)
c
      fes=-beta*ef
c
      return
end
c
      function epa(tt)
c
c     ... computes dielectric loss for alumina
c         interpolates data of Westphal and Sils, p.26
c
      implicit real*8(a-h,o-y)
      real*8 kpts
      dimension tpts(15),kpts(15)
      data tpts/2.5d+01,9.9d+01,1.84d+02,2.81d+02,3.56d+02,4.3d+02
     2,5.62d+02 ,7.05d+02,8.0d+02,9.03d+02,9.73d+02,1.025d+03
     3,1.05d+03,1.109d+03,1.132d+03/
      data kpts/9.31d0,9.41d0,9.58d0,9.72d0,9.84d0,9.96d0,10.17d0
     2,10.42d0,10.63d0,10.86d0,10.98d0,11.17d0,11.22d0,11.38d0,11.41d0/
c
       t=tt-2.73d+02
      if(t.ge.tpts(15))then
          a=(kpts(15)-kpts(14))/(tpts(15)-tpts(14))
          b=kpts(15)-a*tpts(15)
          epa=a*t+b
          return
      endif
      do 5 i=1,14
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
```

```
          epa=kpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          epa=kpts(i)+(t-x)*(kpts(i+1)-kpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function epag(tt)
c
c     ... computes dielectric loss for gamma alumina
c         interpolates data of Mitch Jackson 5/97
c
      implicit real*8(a-h,o-y)
      real*8 kpts
      dimension tpts(12),kpts(12)
      data tpts/2.9d+01,3.5d+01,1.49d+02,2.62d+02,3.71d+02
     2,4.78d+02,5.82d+02,6.84d+02,7.84d+02,8.81d+02,9.76d+02
     3,1.068d+03/
      data kpts/4.487d0,4.498d0,5.040d0,4.448d0,4.652d0
     2,4.917d0,4.647d0,4.441d0,4.313d0,4.196d0,4.046d0
     3,3.717d0/
c
       t=tt-2.73d+02
      if(t.ge.tpts(12))then
          a=(kpts(12)-kpts(11))/(tpts(12)-tpts(11))
          b=kpts(12)-a*tpts(12)
          epag=a*t+b
          return
      endif
```

```
      if(t.le.tpts(1))then
      epag=kpts(1)
      return
      endif
      do 5 i=1,11
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          epag=kpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          epag=kpts(i)+(t-x)*(kpts(i+1)-kpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function eppa(tt)
c
c     ... computes dielectric loss for alumina
c         interpolates data of Westphal and Sils, p.26
c
      implicit real*8(a-h,o-y)
      real*8 kpts
      dimension tpts(15),kpts(15)
      data tpts/2.5d+01,9.9d+01,1.84d+02,2.81d+02,3.56d+02,4.3d+02
     2,5.62d+02,7.05d+02,8.0d+02,9.03d+02,9.73d+02,1.025d+03,1.05d+03
     3,1.109d+03,1.132d+03/
      data kpts/3.63d-03,3.95d-03,5.08d-03,6.8d-03,8.86d-03,1.12d-02
     2,1.63d-02,2.24d-02,2.82d-02,3.58d-02,4.39d-02,5.03d-02,5.61d-02
     3,6.83d-02,1.14d-01/
```

```
c
      t=tt-2.73d+02
      if(t.ge.tpts(15))then
          a=(kpts(15)-kpts(14))/(tpts(15)-tpts(14))
          b=kpts(15)-a*tpts(15)
          eppa=a*t+b
          return
      endif
      do 5 i=1,14
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          eppa=kpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          eppa=kpts(i)+(t-x)*(kpts(i+1)-kpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function eppag(tt)
c
c     ... computes dielectric loss for gamma alumina
c         interpolates data of Mitch Jackson 5/97
c
      implicit real*8(a-h,o-y)
      real*8 kpts
      dimension tpts(12),kpts(12)
      data tpts/2.9d+01,3.5d+01,1.49d+02,2.62d+02,3.71d+02
     2,4.78d+02,5.82d+02,6.84d+02,7.84d+02,8.81d+02,9.76d+02
```

```
     3,1.068d+03/
      data kpts/0.3921d0,0.3893d0,0.2260d0,0.1703d0,0.1523d0
     2,0.1541d0,0.1469d0,0.1547d0,0.1454d0,0.1681d0,0.1921d0
     3,0.1311d0/
c
       t=tt-2.73d+02
      if(t.ge.tpts(12))then
          eppag=0.15d0
          return
      endif
      if(t.le.tpts(1))then
          eppag=kpts(1)
          return
      endif
      do 5 i=1,11
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          eppag=kpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          eppag=kpts(i)+(t-x)*(kpts(i+1)-kpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function eppt(t1,dpt)
c
c     ...13 Oct 95
c
```

```
c        Test using formula (1-59) in Harrington
c
c     ... computes dielectric loss for platinum
c
      implicit double precision(a-h,o-y)
c
      data eps0,pi,freq/8.854d-12,3.141592653589793d0,.915d+09/
c
c     ...approach using resistivity from Shortley & Williams
c
c     Pt:
      data rho0,alph/10.6d-08,3.927d-03/
c     Ag:
c     data rho0,alph/1.6d-08,3.75d-03/
      rho=rho0*(1.d0+alph*(t1-2.93d2))
      sig=1.d0/rho
c
c     ...correct for "quantum dot" effect a la Nimtz, et al
c
      sig=(dpt/5.0d-06)**3*sig
      eppt=sig/(2.d0*pi*freq*eps0)
      return
      end
c
      function cpgas(t)
c
c          29 June 1995
c
c     ... specific heat for air
c
      implicit real*8(a-h,o-y)
      real*8 khes
```

```
      dimension tpts(6),khes(6)
      data tpts/2.d+02,3.d+02,4.d+02,6.d+02,8.d+02,1.d+03/
      data khes/1.007d3,1.007d3,1.014d3,1.051d3,1.099d3,1.141d3/
      if(t.ge.tpts(6))then
          a=(khes(6)-khes(5))/(tpts(6)-tpts(5))
          b=khes(6)-a*tpts(6)
          cpgas=a*t+b
          return
      endif
      do 5 i=1,5
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          cpgas=khes(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          cpgas=khes(i)+(t-x)*(khes(i+1)-khes(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function cgas(t)
c
c          29 June 1995
c
c     ... thermal conductivity for air
c
      implicit real*8(a-h,o-y)
      real*8 khes
      dimension tpts(6),khes(6)
```

```
      data tpts/2.d+02,3.d+02,4.d+02,6.d+02,8.d+02,1.d+03/
      data khes/0.0181d0,0.0263d0,0.0338d0,0.0469d0,0.0573d0,0.0667d0/
      if(t.ge.tpts(6))then
          a=(khes(6)-khes(5))/(tpts(6)-tpts(5))
          b=khes(6)-a*tpts(6)
          cgas=a*t+b
          return
      endif
      do 5 i=1,5
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          cgas=khes(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          cgas=khes(i)+(t-x)*(khes(i+1)-khes(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function conpt(t)
c
c          29 June 1995
c
c     ... thermal conductivity for platinum
c
      implicit real*8(a-h,o-y)
      real*8 kpts
      dimension tpts(9),kpts(9)
      data tpts/2.d+02,3.d+02,4.d+02,6.d+02,8.d+02,1.d+03,1.2d+03
```

```
     2,1.5d+03,2.d+03/
      data kpts/72.6d0,71.6d0,71.8d0,73.2d0,75.6d0,78.7d0,82.6d0,89.5d0,
     2 99.4d0/
      if(t.ge.tpts(9))then
          a=(kpts(9)-kpts(8))/(tpts(9)-tpts(8))
          b=kpts(9)-a*tpts(9)
          conpt=a*t+b
          return
      endif
      do 5 i=1,8
        x=tpts(i)
        y=tpts(i+1)
        if(t.eq.x)then
          conpt=kpts(i)
          return
        elseif(t.gt.x.and.t.lt.y)then
          conpt=kpts(i)+(t-x)*(kpts(i+1)-kpts(i))/(y-x)
          return
        endif
    5 continue
      return
      end
c
      function cons(t,pf)
      implicit real*8(a-h,o-y)
      data refr,ext,sig/1.5d0,3.0d+03,5.669d-08/
      data rho1,rho2/7.d+02,21.45d+03/
      ye1=cal(t)
      ye2=conpt(t)
      vf=pf*rho1/((1.d0-pf)*rho2+pf*rho1)
      yfac=(ye2-ye1)/(2.d0*ye1+ye2)
      yem=(1.d0+2.d0*vf*yfac)/(1.d0-vf*yfac)*ye1
```

```
      cons=yem
c     ks=0.95d0*cal(t)+0.05d0*kpt(t)
      cons=cons+(1.6d1/3.d0)*refr**2*sig*t**3/ext
      return
      end
c
SUBROUTINE DGECO(A,LDA,N,IPVT,RCOND,Z)
      INTEGER LDA,N,IPVT(1)
      DOUBLE PRECISION A(LDA,1),Z(1)
      DOUBLE PRECISION RCOND
      DOUBLE PRECISION DDOT,EK,T,WK,WKM
      DOUBLE PRECISION ANORM,S,DASUM,SM,YNORM
      INTEGER INFO,J,K,KB,KP1,L
      ANORM = 0.0D0
      DO 10 J = 1, N
         ANORM = DMAX1(ANORM,DASUM(N,A(1,J),1))
   10 CONTINUE
      CALL DGEFA(A,LDA,N,IPVT,INFO)
      EK = 1.0D0
      DO 20 J = 1, N
         Z(J) = 0.0D0
   20 CONTINUE
      DO 100 K = 1, N
         IF (Z(K) .NE. 0.0D0) EK = DSIGN(EK,-Z(K))
         IF (DABS(EK-Z(K)) .LE. DABS(A(K,K))) GO TO 30
            S = DABS(A(K,K))/DABS(EK-Z(K))
            CALL DSCAL(N,S,Z,1)
            EK = S*EK
   30    CONTINUE
         WK = EK - Z(K)
         WKM = -EK - Z(K)
         S = DABS(WK)
```

```
      SM = DABS(WKM)
      IF (A(K,K) .EQ. 0.0D0) GO TO 40
         WK = WK/A(K,K)
         WKM = WKM/A(K,K)
      GO TO 50
 40   CONTINUE
         WK = 1.0D0
         WKM = 1.0D0
 50   CONTINUE
      KP1 = K + 1
      IF (KP1 .GT. N) GO TO 90
         DO 60 J = KP1, N
            SM = SM + DABS(Z(J)+WKM*A(K,J))
            Z(J) = Z(J) + WK*A(K,J)
            S = S + DABS(Z(J))
 60      CONTINUE
         IF (S .GE. SM) GO TO 80
            T = WKM - WK
            WK = WKM
            DO 70 J = KP1, N
               Z(J) = Z(J) + T*A(K,J)
 70         CONTINUE
 80      CONTINUE
 90   CONTINUE
      Z(K) = WK
100 CONTINUE
   S = 1.0D0/DASUM(N,Z,1)
   CALL DSCAL(N,S,Z,1)
   DO 120 KB = 1, N
      K = N + 1 - KB
      IF (K .LT. N) Z(K) = Z(K) + DDOT(N-K,A(K+1,K),1,Z(K+1),1)
      IF (DABS(Z(K)) .LE. 1.0D0) GO TO 110
```

```
         S = 1.0D0/DABS(Z(K))
         CALL DSCAL(N,S,Z,1)
110    CONTINUE
       L = IPVT(K)
       T = Z(L)
       Z(L) = Z(K)
       Z(K) = T
120 CONTINUE
    S = 1.0D0/DASUM(N,Z,1)
    CALL DSCAL(N,S,Z,1)
    YNORM = 1.0D0
    DO 140 K = 1, N
       L = IPVT(K)
       T = Z(L)
       Z(L) = Z(K)
       Z(K) = T
       IF (K .LT. N) CALL DAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
       IF (DABS(Z(K)) .LE. 1.0D0) GO TO 130
          S = 1.0D0/DABS(Z(K))
          CALL DSCAL(N,S,Z,1)
          YNORM = S*YNORM
130    CONTINUE
140 CONTINUE
    S = 1.0D0/DASUM(N,Z,1)
    CALL DSCAL(N,S,Z,1)
    YNORM = S*YNORM
    DO 160 KB = 1, N
       K = N + 1 - KB
       IF (DABS(Z(K)) .LE. DABS(A(K,K))) GO TO 150
          S = DABS(A(K,K))/DABS(Z(K))
          CALL DSCAL(N,S,Z,1)
          YNORM = S*YNORM
```

```
150    CONTINUE
       IF (A(K,K) .NE. 0.0D0) Z(K) = Z(K)/A(K,K)
       IF (A(K,K) .EQ. 0.0D0) Z(K) = 1.0D0
       T = -Z(K)
       CALL DAXPY(K-1,T,A(1,K),1,Z(1),1)
160 CONTINUE
    S = 1.0D0/DASUM(N,Z,1)
    CALL DSCAL(N,S,Z,1)
    YNORM = S*YNORM
    IF (ANORM .NE. 0.0D0) RCOND = YNORM/ANORM
    IF (ANORM .EQ. 0.0D0) RCOND = 0.0D0
    RETURN
    END
    SUBROUTINE DGEFA(A,LDA,N,IPVT,INFO)
    INTEGER LDA,N,IPVT(1),INFO
    DOUBLE PRECISION A(LDA,1)
    DOUBLE PRECISION T
    INTEGER IDAMAX,J,K,KP1,L,NM1
    INFO = 0
    NM1 = N - 1
    IF (NM1 .LT. 1) GO TO 70
    DO 60 K = 1, NM1
       KP1 = K + 1
       L = IDAMAX(N-K+1,A(K,K),1) + K - 1
       IPVT(K) = L
       IF (A(L,K) .EQ. 0.0D0) GO TO 40
          IF (L .EQ. K) GO TO 10
             T = A(L,K)
             A(L,K) = A(K,K)
             A(K,K) = T
 10       CONTINUE
          T = -1.0D0/A(K,K)
```

```
         CALL DSCAL(N-K,T,A(K+1,K),1)
         DO 30 J = KP1, N
            T = A(L,J)
            IF (L .EQ. K) GO TO 20
               A(L,J) = A(K,J)
               A(K,J) = T
20          CONTINUE
            CALL DAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)
30       CONTINUE
      GO TO 50
40    CONTINUE
         INFO = K
50    CONTINUE
60 CONTINUE
70 CONTINUE
   IPVT(N) = N
   IF (A(N,N) .EQ. 0.0D0) INFO = N
   RETURN
   END
   SUBROUTINE DAXPY(N,DA,DX,INCX,DY,INCY)
   DOUBLE PRECISION DX(1),DY(1),DA
   INTEGER I,INCX,INCY,IXIY,M,MP1,N
   IF(N.LE.0)RETURN
   IF (DA .EQ. 0.0D0) RETURN
   IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
   IX = 1
   IY = 1
   IF(INCX.LT.0)IX = (-N+1)*INCX + 1
   IF(INCY.LT.0)IY = (-N+1)*INCY + 1
   DO 10 I = 1,N
     DY(IY) = DY(IY) + DA*DX(IX)
     IX = IX + INCX
```

```
      IY = IY + INCY
   10 CONTINUE
      RETURN
   20 M = MOD(N,4)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        DY(I) = DY(I) + DA*DX(I)
   30 CONTINUE
      IF( N .LT. 4 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,4
        DY(I) = DY(I) + DA*DX(I)
        DY(I + 1) = DY(I + 1) + DA*DX(I + 1)
        DY(I + 2) = DY(I + 2) + DA*DX(I + 2)
        DY(I + 3) = DY(I + 3) + DA*DX(I + 3)
   50 CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION DDOT(N,DX,INCX,DY,INCY)
      DOUBLE PRECISION DX(1),DY(1),DTEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
      DDOT = 0.0D0
      DTEMP = 0.0D0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        DTEMP = DTEMP + DX(IX)*DY(IY)
        IX = IX + INCX
```

```
     IY = IY + INCY
10 CONTINUE
   DDOT = DTEMP
   RETURN
20 M = MOD(N,5)
   IF( M .EQ. 0 ) GO TO 40
   DO 30 I = 1,M
     DTEMP = DTEMP + DX(I)*DY(I)
30 CONTINUE
   IF( N .LT. 5 ) GO TO 60
40 MP1 = M + 1
   DO 50 I = MP1,N,5
     DTEMP = DTEMP + DX(I)*DY(I) + DX(I + 1)*DY(I + 1) +
  *   DX(I + 2)*DY(I + 2) + DX(I + 3)*DY(I + 3) + DX(I + 4)*DY(I + 4)
50 CONTINUE
60 DDOT = DTEMP
   RETURN
   END
   SUBROUTINE  DSCAL(N,DA,DX,INCX)
   DOUBLE PRECISION DA,DX(1)
   INTEGER I,INCX,M,MP1,N,NINCX
   IF(N.LE.0)RETURN
   IF(INCX.EQ.1)GO TO 20
   NINCX = N*INCX
   DO 10 I = 1,NINCX,INCX
     DX(I) = DA*DX(I)
10 CONTINUE
   RETURN
20 M = MOD(N,5)
   IF( M .EQ. 0 ) GO TO 40
   DO 30 I = 1,M
     DX(I) = DA*DX(I)
```

```
30 CONTINUE
   IF( N .LT. 5 ) RETURN
40 MP1 = M + 1
   DO 50 I = MP1,N,5
     DX(I) = DA*DX(I)
     DX(I + 1) = DA*DX(I + 1)
     DX(I + 2) = DA*DX(I + 2)
     DX(I + 3) = DA*DX(I + 3)
     DX(I + 4) = DA*DX(I + 4)
50 CONTINUE
   RETURN
   END
   DOUBLE PRECISION FUNCTION DASUM(N,DX,INCX)
   DOUBLE PRECISION DX(1),DTEMP
   INTEGER I,INCX,M,MP1,N,NINCX
   DASUM = 0.0D0
   DTEMP = 0.0D0
   IF(N.LE.0)RETURN
   IF(INCX.EQ.1)GO TO 20
   NINCX = N*INCX
   DO 10 I = 1,NINCX,INCX
     DTEMP = DTEMP + DABS(DX(I))
10 CONTINUE
   DASUM = DTEMP
   RETURN
20 M = MOD(N,6)
   IF( M .EQ. 0 ) GO TO 40
   DO 30 I = 1,M
     DTEMP = DTEMP + DABS(DX(I))
30 CONTINUE
   IF( N .LT. 6 ) GO TO 60
40 MP1 = M + 1
```

```
      DO 50 I = MP1,N,6
        DTEMP = DTEMP + DABS(DX(I)) + DABS(DX(I + 1)) + DABS(DX(I + 2))
     *  + DABS(DX(I + 3)) + DABS(DX(I + 4)) + DABS(DX(I + 5))
   50 CONTINUE
   60 DASUM = DTEMP
      RETURN
      END
      INTEGER FUNCTION IDAMAX(N,DX,INCX)
      DOUBLE PRECISION DX(1),DMAX
      INTEGER I,INCX,IX,N
      IDAMAX = 0
      IF( N .LT. 1 ) RETURN
      IDAMAX = 1
      IF(N.EQ.1)RETURN
      IF(INCX.EQ.1)GO TO 20
      IX = 1
      DMAX = DABS(DX(1))
      IX = IX + INCX
      DO 10 I = 2,N
         IF(DABS(DX(IX)).LE.DMAX) GO TO 5
         IDAMAX = I
         DMAX = DABS(DX(IX))
    5    IX = IX + INCX
   10 CONTINUE
      RETURN
   20 DMAX = DABS(DX(1))
      DO 30 I = 2,N
         IF(DABS(DX(I)).LE.DMAX) GO TO 30
         IDAMAX = I
         DMAX = DABS(DX(I))
   30 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE DGESL(A,LDA,N,IPVT,B,JOB)
      INTEGER LDA,N,IPVT(1),JOB
      DOUBLE PRECISION A(LDA,1),B(1)
      DOUBLE PRECISION DDOT,T
      INTEGER K,KB,L,NM1
      NM1 = N - 1
      IF (JOB .NE. 0) GO TO 50
         IF (NM1 .LT. 1) GO TO 30
         DO 20 K = 1, NM1
            L = IPVT(K)
            T = B(L)
            IF (L .EQ. K) GO TO 10
               B(L) = B(K)
               B(K) = T
10          CONTINUE
            CALL DAXPY(N-K,T,A(K+1,K),1,B(K+1),1)
20       CONTINUE
30       CONTINUE
         DO 40 KB = 1, N
            K = N + 1 - KB
            B(K) = B(K)/A(K,K)
            T = -B(K)
            CALL DAXPY(K-1,T,A(1,K),1,B(1),1)
40       CONTINUE
      GO TO 100
50    CONTINUE
         DO 60 K = 1, N
            T = DDOT(K-1,A(1,K),1,B(1),1)
            B(K) = (B(K) - T)/A(K,K)
60       CONTINUE
         IF (NM1 .LT. 1) GO TO 90
```

```
      DO 80 KB = 1, NM1
         K = N - KB
         B(K) = B(K) + DDOT(N-K,A(K+1,K),1,B(K+1),1)
         L = IPVT(K)
         IF (L .EQ. K) GO TO 70
            T = B(L)
            B(L) = B(K)
            B(K) = T
   70    CONTINUE
   80 CONTINUE
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
      SUBROUTINE LIN(ID,IT,IPVT,U,V,S,E,WORK,A,LDA,N,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(LDA,1),B(1)
      DIMENSION IPVT(1)
      DIMENSION U(LDA,1),V(LDA,1),S(1),E(1),WORK(1)
 1001 FORMAT(25X,'INFO not equal to zero in LIN')
 1002 FORMAT(22X,'Incorrect use of ID and/or IT in LIN')
 1003 FORMAT(10X,I3,1X,'SINGULAR VALUE(S) LESS THAN 10-14 OBTAINED')
      IF((ID.NE.1).AND.(ID.NE.2).AND.(IT.NE.1).AND.(IT.NE.2))THEN
      WRITE(2,1002)
      STOP
      ENDIF
      IF(ID.EQ.2)THEN
      IF(IT.NE.2)THEN
      JOB=11
      CALL DSVDC(A,LDA,N,N,S,E,U,LDA,V,LDA,WORK,JOB,INFO)
      IF(INFO.NE.0)THEN
        WRITE(2,1001)
```

```
      STOP
    ENDIF
    INDEX=0
    VVV=S(1)
    DO 10 I=1,N
      CN=S(I)/VVV
      IF(CN.LT.1.D-14)THEN
    INDEX=INDEX+1
    S(I)=0.D0
    IF(I.EQ.N)WRITE(2,1003)INDEX
      ELSE
    S(I)=1.D0/S(I)
      ENDIF
10    CONTINUE
    IF(INDEX.GT.5) STOP
    ENDIF
    DO 20 I=1,N
    WORK(I)=0.D0
20  CONTINUE
    DO 40 I=1,N
    DO 30 J=1,N
      WORK(I)=WORK(I)+U(J,I)*B(J)*S(I)
30     CONTINUE
40   CONTINUE
    DO 50 I=1,N
    B(I)=0.D0
50  CONTINUE
    DO 70 J=1,N
    DO 60 I=1,N
      B(I)=B(I)+V(I,J)*WORK(J)
60     CONTINUE
70   CONTINUE
```

```
 ELSE
 JOB=0
 IF(IT.EQ.1)THEN
 CALL DGEFA(A,LDA,N,IPVT,INFO)
 CALL DGESL(A,LDA,N,IPVT,B,JOB)
 ELSE
 CALL DGESL(A,LDA,N,IPVT,B,JOB)
 ENDIF
 ENDIF
 END
 SUBROUTINE DSVDC(X,LDX,N,P,S,E,U,LDU,V,LDV,WORK,JOB,INFO)
 INTEGER LDX,N,P,LDU,LDV,JOB,INFO
 DOUBLE PRECISION X(LDX,1),S(1),E(1),U(LDU,1),V(LDV,1),WORK(1)
 INTEGER I,ITER,J,JOBU,K,KASE,KK,L,LL,LLS,LM1,LP1,LS,LU,M,MAXIT,
*        MM,MM1,MP1,NCT,NCTP1,NCU,NRT,NRTP1
 DOUBLE PRECISION DDOT,T,R
 DOUBLE PRECISION B,C,CS,EL,EMM1,F,G,DNRM2,SCALE,SHIFT,SL,SM,SN,
*                 SMM1,T1,TEST,ZTEST
 LOGICAL WANTU,WANTV
 MAXIT = 50
 WANTU = .FALSE.
 WANTV = .FALSE.
 JOBU = MOD(JOB,100)/10
 NCU = N
 IF (JOBU .GT. 1) NCU = MIN0(N,P)
 IF (JOBU .NE. 0) WANTU = .TRUE.
 IF (MOD(JOB,10) .NE. 0) WANTV = .TRUE.
 INFO = 0

 NCT = MIN0(N-1,P)

 NRT = MAX0(0,MIN0(P-2,N))
```

```
      LU = MAX0(NCT,NRT)
      IF (LU .LT. 1) GO TO 170
      DO 160 L = 1, LU
         LP1 = L + 1
         IF (L .GT. NCT) GO TO 20
            S(L) = DNRM2(N-L+1,X(L,L),1)
            IF (S(L) .EQ. 0.0D0) GO TO 10
               IF (X(L,L) .NE. 0.0D0) S(L) = DSIGN(S(L),X(L,L))
               CALL DSCAL(N-L+1,1.0D0/S(L),X(L,L),1)
               X(L,L) = 1.0D0 + X(L,L)
10          CONTINUE
            S(L) = -S(L)
20       CONTINUE
         IF (P .LT. LP1) GO TO 50
         DO 40 J = LP1, P
            IF (L .GT. NCT) GO TO 30
```

```
         IF (S(L) .EQ. 0.0D0) GO TO 30
            T = -DDOT(N-L+1,X(L,L),1,X(L,J),1)/X(L,L)
            CALL DAXPY(N-L+1,T,X(L,L),1,X(L,J),1)
30       CONTINUE
         E(J) = X(L,J)
40    CONTINUE
50    CONTINUE
      IF (.NOT.WANTU .OR. L .GT. NCT) GO TO 70
         DO 60 I = L, N
            U(I,L) = X(I,L)
60       CONTINUE
70    CONTINUE
      IF (L .GT. NRT) GO TO 150
         E(L) = DNRM2(P-L,E(LP1),1)
         IF (E(L) .EQ. 0.0D0) GO TO 80
            IF (E(LP1) .NE. 0.0D0) E(L) = DSIGN(E(L),E(LP1))
```

```
            CALL DSCAL(P-L,1.0D0/E(L),E(LP1),1)
            E(LP1) = 1.0D0 + E(LP1)
 80      CONTINUE
         E(L) = -E(L)

         IF (LP1 .GT. N .OR. E(L) .EQ. 0.0D0) GO TO 120
            DO 90 I = LP1, N
               WORK(I) = 0.0D0
 90         CONTINUE
            DO 100 J = LP1, P
               CALL DAXPY(N-L,E(J),X(LP1,J),1,WORK(LP1),1)
100         CONTINUE
            DO 110 J = LP1, P
               CALL DAXPY(N-L,-E(J)/E(LP1),WORK(LP1),1,X(LP1,J),1)
110         CONTINUE
120      CONTINUE
```

```
          IF (.NOT.WANTV) GO TO 140
             DO 130 I = LP1, P
                V(I,L) = E(I)
130          CONTINUE
140       CONTINUE
150    CONTINUE
160 CONTINUE
170 CONTINUE
    M = MIN0(P,N+1)
    NCTP1 = NCT + 1
    NRTP1 = NRT + 1
    IF (NCT .LT. P) S(NCTP1) = X(NCTP1,NCTP1)
    IF (N .LT. M) S(M) = 0.0D0
    IF (NRTP1 .LT. M) E(NRTP1) = X(NRTP1,M)
    E(M) = 0.0D0
    IF (.NOT.WANTU) GO TO 300
```

```
      IF (NCU .LT. NCTP1) GO TO 200
      DO 190 J = NCTP1, NCU
         DO 180 I = 1, N
            U(I,J) = 0.0D0
180      CONTINUE
         U(J,J) = 1.0D0
190   CONTINUE
200   CONTINUE
      IF (NCT .LT. 1) GO TO 290
      DO 280 LL = 1, NCT
         L = NCT - LL + 1
         IF (S(L) .EQ. 0.0D0) GO TO 250
            LP1 = L + 1
            IF (NCU .LT. LP1) GO TO 220
            DO 210 J = LP1, NCU
               T = -DDOT(N-L+1,U(L,L),1,U(L,J),1)/U(L,L)
```

```
               CALL DAXPY(N-L+1,T,U(L,L),1,U(L,J),1)

210            CONTINUE

220            CONTINUE

               CALL DSCAL(N-L+1,-1.0D0,U(L,L),1)

               U(L,L) = 1.0D0 + U(L,L)

               LM1 = L - 1

               IF (LM1 .LT. 1) GO TO 240

               DO 230 I = 1, LM1

                  U(I,L) = 0.0D0

230            CONTINUE

240            CONTINUE

            GO TO 270

250         CONTINUE

               DO 260 I = 1, N

                  U(I,L) = 0.0D0

260            CONTINUE
```

```
            U(L,L) = 1.0D0

270       CONTINUE

280    CONTINUE

290    CONTINUE



300 CONTINUE

    IF (.NOT.WANTV) GO TO 350

       DO 340 LL = 1, P

          L = P - LL + 1

          LP1 = L + 1

          IF (L .GT. NRT) GO TO 320

          IF (E(L) .EQ. 0.0D0) GO TO 320

             DO 310 J = LP1, P

                T = -DDOT(P-L,V(LP1,L),1,V(LP1,J),1)/V(LP1,L)

                CALL DAXPY(P-L,T,V(LP1,L),1,V(LP1,J),1)

310          CONTINUE
```

```
320       CONTINUE

          DO 330 I = 1, P

             V(I,L) = 0.0D0

330       CONTINUE

          V(L,L) = 1.0D0

340    CONTINUE

350 CONTINUE

    MM = M

    ITER = 0

360 CONTINUE

       IF (M .EQ. 0) GO TO 620

       IF (ITER .LT. MAXIT) GO TO 370

          INFO = M

          GO TO 620

370    CONTINUE

       DO 390 LL = 1, M
```

```
            L = M - LL
            IF (L .EQ. 0) GO TO 400
            TEST = DABS(S(L)) + DABS(S(L+1))
            ZTEST = TEST + DABS(E(L))
            IF (ZTEST .NE. TEST) GO TO 380
               E(L) = 0.0D0
               GO TO 400
380         CONTINUE
390      CONTINUE
400      CONTINUE
         IF (L .NE. M - 1) GO TO 410
            KASE = 4
         GO TO 480
410      CONTINUE
            LP1 = L + 1
            MP1 = M + 1
```

```
      DO 430 LLS = LP1, MP1
         LS = M - LLS + LP1
         IF (LS .EQ. L) GO TO 440
         TEST = 0.0D0
         IF (LS .NE. M) TEST = TEST + DABS(E(LS))
         IF (LS .NE. L + 1) TEST = TEST + DABS(E(LS-1))
         ZTEST = TEST + DABS(S(LS))
         IF (ZTEST .NE. TEST) GO TO 420
            S(LS) = 0.0D0
            GO TO 440
420      CONTINUE
430   CONTINUE
440   CONTINUE
      IF (LS .NE. L) GO TO 450
         KASE = 3
      GO TO 470
```

```
450       CONTINUE
          IF (LS .NE. M) GO TO 460
             KASE = 1
          GO TO 470
460       CONTINUE
             KASE = 2
             L = LS
470       CONTINUE
480    CONTINUE
       L = L + 1
       GO TO (490,520,540,570), KASE
490    CONTINUE
          MM1 = M - 1
          F = E(M-1)
          E(M-1) = 0.0D0
          DO 510 KK = L, MM1
```

```
            K = MM1 - KK + L

            T1 = S(K)

            CALL DROTG(T1,F,CS,SN)

            S(K) = T1

            IF (K .EQ. L) GO TO 500

               F = -SN*E(K-1)

               E(K-1) = CS*E(K-1)

500         CONTINUE

            IF (WANTV) CALL DROT(P,V(1,K),1,V(1,M),1,CS,SN)

510      CONTINUE

      GO TO 610

520   CONTINUE

         F = E(L-1)

         E(L-1) = 0.0D0

         DO 530 K = L, M

            T1 = S(K)
```

```
            CALL DROTG(T1,F,CS,SN)

            S(K) = T1

            F = -SN*E(K)

            E(K) = CS*E(K)

            IF (WANTU) CALL DROT(N,U(1,K),1,U(1,L-1),1,CS,SN)

530         CONTINUE

        GO TO 610

540     CONTINUE

            SCALE = DMAX1(DABS(S(M)),DABS(S(M-1)),DABS(E(M-1)),

     *                    DABS(S(L)),DABS(E(L)))

            SM = S(M)/SCALE

            SMM1 = S(M-1)/SCALE

            EMM1 = E(M-1)/SCALE

            SL = S(L)/SCALE

            EL = E(L)/SCALE

            B = ((SMM1 + SM)*(SMM1 - SM) + EMM1**2)/2.0D0
```

```
      C = (SM*EMM1)**2

      SHIFT = 0.0D0

      IF (B .EQ. 0.0D0 .AND. C .EQ. 0.0D0) GO TO 550

         SHIFT = DSQRT(B**2+C)

         IF (B .LT. 0.0D0) SHIFT = -SHIFT

         SHIFT = C/(B + SHIFT)

550   CONTINUE

      F = (SL + SM)*(SL - SM) - SHIFT

      G = SL*EL

      MM1 = M - 1

      DO 560 K = L, MM1

         CALL DROTG(F,G,CS,SN)

         IF (K .NE. L) E(K-1) = F

         F = CS*S(K) + SN*E(K)

         E(K) = CS*E(K) - SN*S(K)

         G = SN*S(K+1)
```

```
            S(K+1) = CS*S(K+1)

            IF (WANTV) CALL DROT(P,V(1,K),1,V(1,K+1),1,CS,SN)

            CALL DROTG(F,G,CS,SN)

            S(K) = F

            F = CS*E(K) + SN*S(K+1)

            S(K+1) = -SN*E(K) + CS*S(K+1)

            G = SN*E(K+1)

            E(K+1) = CS*E(K+1)

            IF (WANTU .AND. K .LT. N)

     *         CALL DROT(N,U(1,K),1,U(1,K+1),1,CS,SN)

  560    CONTINUE

         E(M-1) = F

         ITER = ITER + 1

      GO TO 610

  570 CONTINUE

         IF (S(L) .GE. 0.0D0) GO TO 580
```

```
            S(L) = -S(L)
            IF (WANTV) CALL DSCAL(P,-1.0D0,V(1,L),1)
580       CONTINUE
590       IF (L .EQ. MM) GO TO 600
            IF (S(L) .GE. S(L+1)) GO TO 600
            T = S(L)
            S(L) = S(L+1)
            S(L+1) = T
            IF (WANTV .AND. L .LT. P)
     *         CALL DSWAP(P,V(1,L),1,V(1,L+1),1)
            IF (WANTU .AND. L .LT. N)
     *         CALL DSWAP(N,U(1,L),1,U(1,L+1),1)
            L = L + 1
          GO TO 590
600       CONTINUE
          ITER = 0
```

```
            M = M - 1
  610    CONTINUE
      GO TO 360
  620 CONTINUE
      RETURN
      END
c
      SUBROUTINE  DSWAP (N,DX,INCX,DY,INCY)
      DOUBLE PRECISION DX(1),DY(1),DTEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
```

```
      DO 10 I = 1,N
        DTEMP = DX(IX)
        DX(IX) = DY(IY)
        DY(IY) = DTEMP
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
   20 M = MOD(N,3)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        DTEMP = DX(I)
        DX(I) = DY(I)
        DY(I) = DTEMP
   30 CONTINUE
      IF( N .LT. 3 ) RETURN
```

```
   40 MP1 = M + 1

      DO 50 I = MP1,N,3

        DTEMP = DX(I)

        DX(I) = DY(I)

        DY(I) = DTEMP

        DTEMP = DX(I + 1)

        DX(I + 1) = DY(I + 1)

        DY(I + 1) = DTEMP

        DTEMP = DX(I + 2)

        DX(I + 2) = DY(I + 2)

        DY(I + 2) = DTEMP

   50 CONTINUE

      RETURN

      END

c

      SUBROUTINE DROTG(DA,DB,C,S)
```

```
      DOUBLE PRECISION DA,DB,C,S,ROE,SCALE,R,Z
      ROE = DB
      IF( DABS(DA) .GT. DABS(DB) ) ROE = DA
      SCALE = DABS(DA) + DABS(DB)
      IF( SCALE .NE. 0.0D0 ) GO TO 10
         C = 1.0D0
         S = 0.0D0
         R = 0.0D0
         GO TO 20
   10 R = SCALE*DSQRT((DA/SCALE)**2 + (DB/SCALE)**2)
      R = DSIGN(1.0D0,ROE)*R
      C = DA/R
      S = DB/R
   20 Z = 1.0D0
      IF( DABS(DA) .GT. DABS(DB) ) Z = S
      DA = R
```

```
      DB = Z

      RETURN

      END

c

     DOUBLE PRECISION FUNCTION DNRM2 ( N, DX, INCX)

      INTEGER       NEXT

      DOUBLE PRECISION DX(1), CUTLO, CUTHI, HITEST, SUM, XMAX,ZERO,ONE

      DATA   ZERO, ONE /0.0D0, 1.0D0/

      DATA CUTLO, CUTHI / 8.232D-11,  1.304D19 /

      IF(N .GT. 0) GO TO 10

      DNRM2= ZERO

      GO TO 300

   10 ASSIGN 30 TO NEXT

      SUM = ZERO

      NN = N * INCX

      I = 1
```

```
 20  GO TO NEXT,(30, 50, 70, 110)
 30 IF( DABS(DX(I)) .GT. CUTLO) GO TO 85
    ASSIGN 50 TO NEXT
    XMAX = ZERO
 50 IF( DX(I) .EQ. ZERO) GO TO 200
    IF( DABS(DX(I)) .GT. CUTLO) GO TO 85
    ASSIGN 70 TO NEXT
    GO TO 105
100 I = J
    ASSIGN 110 TO NEXT
    SUM = (SUM / DX(I)) / DX(I)
105 XMAX = DABS(DX(I))
    GO TO 115
 70 IF( DABS(DX(I)) .GT. CUTLO ) GO TO 75
110 IF( DABS(DX(I)) .LE. XMAX ) GO TO 115
    SUM = ONE + SUM * (XMAX / DX(I))**2
```

```
      XMAX = DABS(DX(I))
      GO TO 200
  115 SUM = SUM + (DX(I)/XMAX)**2
      GO TO 200
   75 SUM = (SUM * XMAX) * XMAX
   85 HITEST = CUTHI/FLOAT( N )
      DO 95 J =I,NN,INCX
      IF(DABS(DX(J)) .GE. HITEST) GO TO 100
   95  SUM = SUM + DX(J)**2
      DNRM2 = DSQRT( SUM )
      GO TO 300
  200 CONTINUE
      I = I + INCX
      IF ( I .LE. NN ) GO TO 20
      DNRM2 = XMAX * DSQRT(SUM)
  300 CONTINUE
```

```
      RETURN
      END
c
      SUBROUTINE  DROT (N,DX,INCX,DY,INCY,C,S)
C
C     APPLIES A PLANE ROTATION.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      DOUBLE PRECISION DX(1),DY(1),DTEMP,C,S
      INTEGER I,INCX,INCY,IX,IY,N
C
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C       CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS NOT EQUAL
C         TO 1
```

```
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        DTEMP = C*DX(IX) + S*DY(IY)
        DY(IY) = C*DY(IY) - S*DX(IX)
        DX(IX) = DTEMP
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
C       CODE FOR BOTH INCREMENTS EQUAL TO 1
C
```

```
20 DO 30 I = 1,N

     DTEMP = C*DX(I) + S*DY(I)

     DY(I) = C*DY(I) - S*DX(I)

     DX(I) = DTEMP

30 CONTINUE

   RETURN

   END
```

Vita

Max Savransky was born in Ukraine Kiev, on November 16, 1971. In June of 1979, Max and his family arrived in New York City after spending a two weeks in Austria and three months in Italy. In 1987, Max began in Fair Lawn High School in New Jersey, and in June of 1990 he received his high school diploma. In the fall of 1990, Max began his undergraduate studies at The New Jersey Institute of Technology, and in the spring of 1994 he received his Bachelor of Science degree in Mechanical Engineering. In the Fall of 1994, Max began his graduate studies in at The Virginia Polytechnic Institute and State University. He completed his Master of Science degree in Mechanical Engineering in December 1996. He then continued his studies as a doctoral student while also pursuing the Master of Science degree in Mathematics which he completed in May 2003. He completed his Ph.D in Mechanical Engineering in December of 2003.

Permanent Address: 36-03 Stelton Terrace
Fair Lawn, NJ 07410

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Greg Walker, Department of Mechanical Engineering, Virginia Tech.