OPTIMIZATION OF PROCESS PARAMETERS OF FLUIDIZED BED BIOMASS COMBUSTOR USING CFD MODELING

Dalthota Gedara Chathuranga Wickramasinghe

178034K

Degree of Master of Science

Department of Chemical & Process Engineering

University of Moratuwa

Sri Lanka

October 2018

OPTIMIZATION OF PROCESS PARAMETERS OF FLUIDIZED BED BIOMASS COMBUSTOR USING CFD MODELING

Dalthota Gedara Chathuranga Wickramasinghe

178034K

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science in fulltime research in Chemical & Process Engineering

Department of Chemical & Process Engineering

University of Moratuwa

Sri Lanka

October 2018

Declaration

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Signature: …… Date: ……..

Copyright Statement

In addition, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …… Date: ……..

Declaration (Supervisor)

The above candidate has carried out research for the Master thesis/dissertation under my supervision and I accepted this thesis/dissertation for degree award.

Signature of the Supervisor: …… Date: ……..

Dr. M. Narayana

Senior Lecturer

Department of Chemical & Process Engineering

University of Moratuwa

Signature of the Co-Supervisor: …… Date: ……..

Prof. A. D. U. S. Amarasinghe

Professor

Department of Chemical & Process Engineering

University of Moratuwa

Abstract

The present work is focused on mathematical modelling of fluidized bed combustion of biomass by means of computational fluid dynamics (CFD). Introducing a stable CFD solver for combustion of solid fuel with Large-Eddy Simulation (LES) turbulence modeling and diffusion corrective velocity is the novelty of this study. A great part of the work is devoted to development of a reliable algorithm to solve the fluidized bed combustion model and to improve the accuracy of simulation results. A three-dimensional transient model is developed with a special emphasis on the conservativeness property of energy and mass.

The second part of the work is devoted to development of C++ libraries for the open source CFD tool kit OpenFOAM for coupling fluidized bed model with gas phase combustion. The created CFD solver has capability to adept with wide range of application of fluidized bed combustion. The presented results contribute to better understanding of numerical modelling and simulation of fluidized bed combustion, especially optimizing of combustor geometries.

The performance of fluidized bed biomass combustor depends on the turbulence of freeboard. The optimum amount of secondary air ratio is 1.6%, which gives higher freeboard temperature while maintaining minimum emission. The optimum excess air ratio is 10% for 0.1125 kg/h. In the case of suspension combustion, maximum combustor temperature has been obtained for optimum airflow, which was 5.5 ms^{-1} of inlet air velocity for a 0.00171 kg/s of fuel feeding rate (65% amount of excess air).

Keywords

Computational Fluid Dynamics, fluidized bed, biomass, Large-Eddy Simulation, Combustion

Acknowledgements

I am thankful to my supervisor Dr. M. Narayana and co-supervisor Prof. A. D. U. S. Amarasinghe for their encouragement, guidance and time throughout my study. I am also thankful to Dr. S. H. P. Gunawardena, Head of the Department of Chemical & Process Engineering for allowing me to study for my M.Sc. in the department.

Finally, I should thank laboratory staff of Department of Chemical & Process Engineering, University of Moratuwa and my colleagues for their supports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

C_d	Drag coefficient
D_i	Effective mass diffusion coefficient for species *i* (m^2/s)
d	Particle diameter (m)
f_g	Drag force acting on particle (N)
f_p	Contact force acting on particle (N)
g	Gravitational acceleration (m/s^2)
h_s	Sensible enthalpy of gas phase (J/kg)
m_p	Particle mass (kg)
P	Pressure (Pa)
Re	Reynolds number
S_h	Enthalpy source term due to homogeneous reaction (W/m^3)
$S_{p,h}$	Enthalpy source term from particle phase (W/m^3)
$S_{p,mom}$	Momentum source term (N/m^3)
$S_{p,m}$	Mass source term ($kgm^{-3}s^{-1}$)
S_{p,Y_i}	Species source term from particle phase ($kgm^{-3}s^{-1}$)
S_{rad}	Radiation source term (W/m^3)
S_{Y_i}	Species source term due to gas phase reactions ($kgm^{-3}s^{-1}$)
T	Temperature (K)
u_g	Gas velocity (m/s)
v_p	Particle velocity (m/s)
V_p	Particle volume (m^3)
V_c	Correction velocity (m/s)
Y_i	Mass fraction of species *i*
ε_g	Volume fraction of gas
ρ_g	Gas density (kg/m^3)
τ_{eff}	Effective stress tenor (Pa)
α_{eff}	Effective thermal diffusivity ($kgm^{-1}s^{-1}$)
β	Inter-phase momentum exchange coefficient ($kgm^{-3}s^{-1}$)
μ	Viscosity ($kgm^{-1}s^{-1}$)

LIST OF APPENDICES

1. INTRODUCTION

During the past few decades, fluidized bed biomass combustors have received significant attention by not only researchers involved in combustion science but also various industries. It is because of not only renewable and environmentally friendly biomass has been found as a promising alternative to fossil fuels but also fluidized bed incineration furnaces have become a widely used method of power generation and solid waste incineration.

Many experimental measurements have been carried out in order to improve the effectiveness of energy utilization as well as to reduce the pollutant emissions and undesirable effects of combustion products on heat transfer surfaces. However, due to the difficulties and expensiveness of experiments, the research has also been focused on development of a reliable computational model as a supporting tool for design and optimization of combustion systems.

1.1 CFD modeling

CFD modelling has become a standard supporting tool for design and optimization of many kinds of combustion systems. The area of applications ranges from small-scale units such as domestic wood stoves and biomass pellet boilers [1], [2], to large scale units such as dryers, gasifiers, fluidized bed and pulverized-fuel furnaces as well as municipal solid waste (MSW) grate and rotary kiln incinerators [3], [4].

The mathematical modeling of gas flow in fluidized beds began in the 60s with Davidson and Harrison [5], where they analyzed mainly bubble motion with bubble diameter, system instability and mass transfer with diffusion and convective contribution. The model was derived from the Kinetic Theory of Granular Flows (KTGF). After that, some numerical models were developed, such as Discrete Particle Model (DPM), the Two Fluid Model (TFM) and Multiphase Particle-In-Cell (MPPIC) [6], [7], with the aim of developing greater control of the system.

There are several mathematical correlations, which describe the fluidized bed process, but these models depend on the application [6]. For a very precise description of the

process, one has to analyze a complex set of equations before starting the solving procedure [8]. The hydrodynamic models describe the motion and distribution of solids, gas–solid mixture, size, velocity and mass and heat transfer phenomena [9]. The gas–solid inter phase model is generally a two-phase model [5], [10].

Kuipers et al. [44] numerically analyzed the hydrodynamics of the fluidized bed, applying the finite differences method. Van Lare [11] studied the influence of particle size on mass transfer in a fluidized bed by using two-fluid model. Carvalho [8] studied methanol production in biomass fluidized bed. The author also claimed that improving the hydrodynamic description is necessary, especially for solid phase. Matos [8] studied coke combustion in a fluidized bed, based on two-fluid model. Tarelho [8] studied the control of gaseous emissions during coal combustion in a fluidized bed. CFD tools have also been used to model the dynamics of pulverized coal and biomass co-firing [12].

It can be concluded that current CFD codes such as OpenFOAM [13] have become very practical tools for analysis of combustion systems. However, quality of simulation results is strongly affected by the quality of input data. Therefore, in order to develop a reliable CFD model, care must be taken when preparing the mesh and defining not only physical and chemical models, but also boundary conditions, which are sometimes necessary to incorporate into the CFD code to adjust the solution procedure and enhance capabilities of the CFD modelling tool [14].

1.2 Objectives and scope of the thesis

This study intends to develop a three dimensional Computational Fluid Dynamics model to analyze fluidized bed combustion of biomass, which can demonstrate the effect of turbulence in the freeboard in order to find optimum process parameters such as particle size of biomass and excess air ratio. Separate freeboard analysis is not sustained alone since it cannot be validated due to difficulties of gathering required measurements (gas composition and temperature profile of the freeboard boundary). Therefore, simultaneous bed analysis also has to be done to overcome this difficulty. The complete model will predict the freeboard temperature gradient and flue gas concentration in the means of time-dependent variations and spatial- dependent

variations. Finally, we can optimize the secondary air ratio as well as air-inject positions in order to achieve improved performance.

Objectives:

- To develop a CFD model for the fluidized bed biomass combustion to demonstrate the effect of turbulence in the freeboard based on Large Eddy Simulation (LES) algorithms and validate experimentally.
- To utilize the results of CFD mathematical model in order to optimize the process parameters for efficient combustion and minimized emission.

1.3 Report structure

The thesis is organized as follows:

Chapter 2 presents mathematical modeling in an organized way. Firstly, mathematical framework is described. Then, governing equations and modeling of heat transfer are presented. The modelling assumptions with various source term modeling are presented afterword including the gas phase chemistry. Furthermore, Chapter 2 summarizes physical and chemical properties of biomass and reaction rate constants.

The discretization of governing equations, are described in the Chapter 3. Especially, convergence and accuracy of results are discussed here. The OpenFOAM case development is introduced in Chapter 3. Model validation cases are presented in Chapter 4, which illustrate the prediction accuracy of proposed CFD solver.

Results are presented in Chapter 5 with especial case study related to suspension combustion. Conclusions are made in the Chapter 6 with suggestions for future works.

2. MATHEMATICAL MODELING

This section describes mathematical modeling systematically. Firstly, the mathematical framework, which is the foundation of final CFD solver, is presented. A general description of modeling of particle motion is provided and two different techniques are described. After that, governing equations of fluid phase are described followed by source term modeling. Finally, much important turbulence modeling is described.

2.1 Computational approach

Generally, CFD models can be categorized into two computational approaches (frameworks): Eulerian–Eulerian and Eulerian–Lagrangian as shown in Figure 1. In Eulerian–Eulerian approach, both particle and gas phases are treated as separated continuous phases. Same sets of governing equations are solved in both phases. The interface transport phenomena have been modeled separately. However, it can only predict the macroscopic characteristics of the particle phase due to continuous modeling approach. This disadvantage can be overcome by Eulerian–Lagrangian approach [15] [16], which treats the particle phase in a discrete manner.

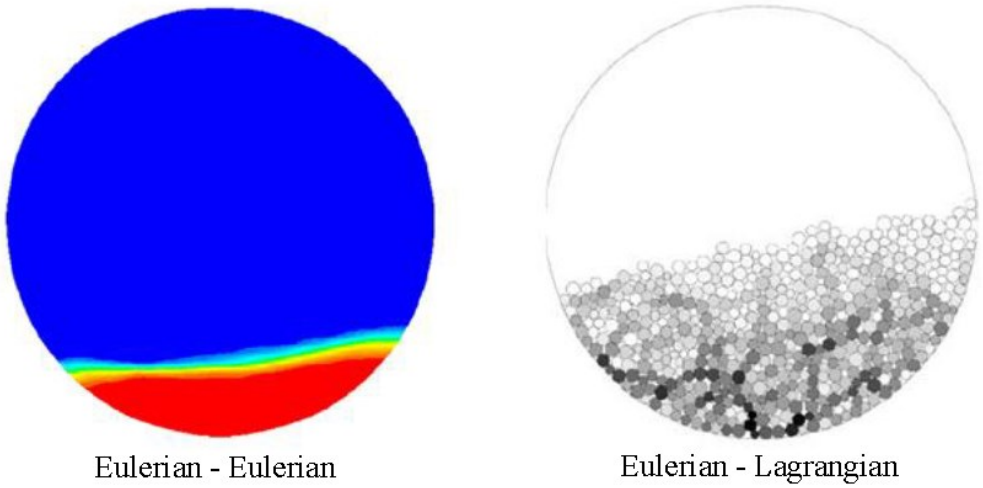

Figure 1: Two computational approaches

Source: K. Jang, W. Han, and K. Y. Huh, "Simulation of moving-bed and fluidized-bed reactors by DPM and MPPIC in OpenFOAM," in 11th OpenFOAM Workshop, 2013. [4]

2.1 Particle motion (Lagrangian phase)

There are two main computational methods for handling Lagrangian particles in OpenFOAM, DEM (Discrete Element Method) [17] and MPPIC (Multiphase Particle-In-Cell) [18]. The DEM requires large computational resources to track the actual number of particles with all collisions between any pair of particles taken into account. It is practically impossible to apply the DEM to industrial devices with reasonable amount of particles. The DPM is a simplified version of DEM with a reduced number of computational parcels composed of the particles of the same characteristics. MPPIC also involves less computational load than DEM by calculating for interaction between the particles on the Eulerian grid. Particle properties are interpolated to the grid and flow field is updated on grid.

The particle motion is governed by following equations (2.1 – 2.4) based on Newton's laws of motion. The total packing (stress) force (f_p) is calculated by Harris and Crighton packing model [19], [20], [7].

$$m_p \frac{dv_p}{dt} = f_g + f_p + m_p g \tag{2.1}$$

$$f_g = \left({V_p \beta}/{1 - \varepsilon_g}\right)(u_g - v_p) \tag{2.2}$$

$$\beta = \begin{cases} 150\left(\left(1-\varepsilon_g\right)^2 \mu / {\varepsilon_g}^2 d^2\right) + 1.75\left(\left(1-\varepsilon_g\right)\rho / \varepsilon_g d\right)\left|u_g - v_p\right| \;\; for\ \varepsilon_g < 0.8 \\ 0.75 C_d \left(\left(1-\varepsilon_g\right)\rho / d\right)\left|u_g - v_p\right| {\varepsilon_g}^{-2.65} \;\; for\ \varepsilon_g \geq 0.8 \end{cases}$$

(2.3)

$$C_d = \begin{cases} \left(24/Re_p\right)\left(1 + 0.15 Re_p^{0.687}\right) \;\; for\ Re_p < 1000 \\ 0.44 \;\; for\ Re_p \geq 1000 \end{cases} \tag{2.4}$$

2.2 Governing equations

A generalized governing equation for variable ϕ, based on the Navier-Stock can be presented by Equation 2.5. Effect of Lagrangian phase has been included as fluid volume fraction (ε_g) in generalized governing equation [21].

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g \phi\right) + \nabla \cdot \left(\varepsilon_g u_g \rho_g \phi\right) = \nabla \cdot \left(\varepsilon_g \rho_g D \nabla \phi\right) + S_\phi \quad (2.5)$$

Where,

$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g \phi\right)$ - Rate of increase of ϕ in control volume

$\nabla \cdot \left(\varepsilon_g u_g \rho_g \phi\right)$ - Net convection of ϕ out from control volume

$\nabla \cdot \left(\varepsilon_g \rho_g D \nabla \phi\right)$ - Net diffusion of ϕ into control volume

S_ϕ - Net generation of ϕ within control volume

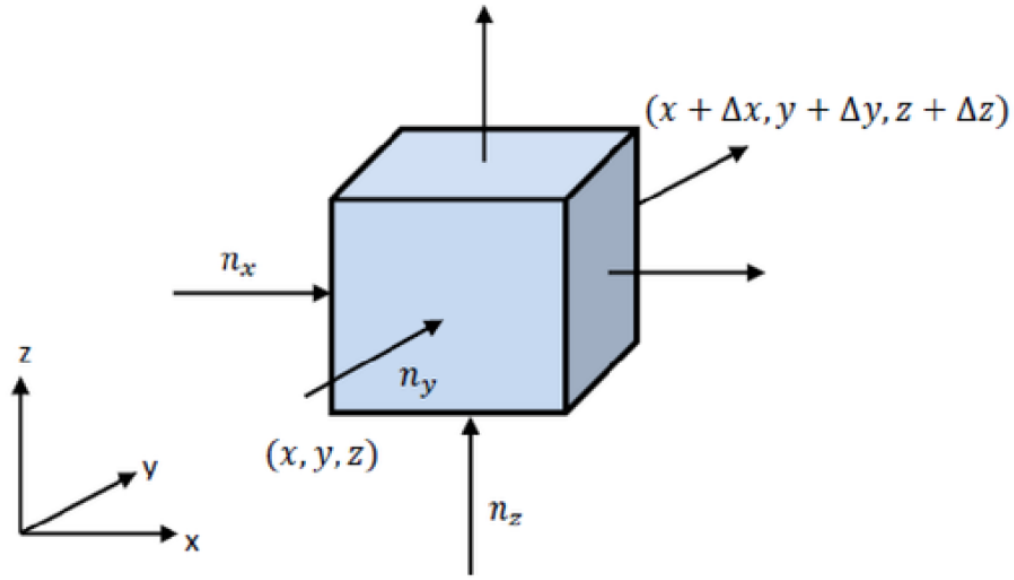

Figure 2: Differential volume element (cell) and fluxes across its faces

2.2.1 Momentum conservation

The momentum conservation for a control volume (Figure 2) can be presented by Equation 2.6. In this equation source term (net momentum generation) can be divided into two categorizes namely, momentum generation due to Lagrangian phase and momentum generation due to external forces that are acting on control volume. However, both categories are calculated based on Newton's laws. The effect of turbulence has been included in the diffusion term. The τ_{eff} in diffusion term is

effective stress tensor, which is based on, not only turbulent viscosity but also laminar viscosity.

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g u_g\right) + \nabla \cdot \left(\varepsilon_g \rho_g u_g u_g\right) = -\nabla p + \nabla \cdot \left(\varepsilon_g \tau_{eff}\right) + \varepsilon_g \rho_g g + S_{p,mom} \quad (2.6)$$

2.2.2 Mass conservation

There are two types of transport equations. One type is total mass conservation that is given by Equation 2.7.

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g\right) + \nabla \cdot \left(\varepsilon_g \rho_g u_g\right) = S_{p,m} \quad (2.7)$$

The second type is species transport equation that is given by Equation 2.8, where $\sum_i Y_i = 1$.

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g Y_i\right) + \nabla \cdot \left(\varepsilon_g (u_g + V_i) \rho_g Y_i\right) = S_{Y_i} + S_{p,Y_i} \quad (2.8)$$

The term V_i represents the diffusion velocity of specie *i*. Finding of actual diffusion velocities for all species in a 3-dimensional space is expensive in the manner of numerical calculation. Therefore, diffusion velocity is approximated by Equation 2.9, which is based on Fick's law.

$$V_i = -D_i \left({}^{\nabla Y_i}/_{Y_i}\right) \quad (2.9)$$

Here, D_i is an effective diffusion coefficient of specie *i*. Now, Equation 2.8 can be rewritten as Equation 2.10.

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g Y_i\right) + \nabla \cdot \left(\varepsilon_g u_g \rho_g Y_i\right) = \nabla \cdot \left(\varepsilon_g \rho_g D_i \nabla Y_i\right) + S_{Y_i} + S_{p,Y_i} \quad (2.10)$$

However, due to the above approximation for diffusion velocity, there will be new issue relating to the mass conservation. In order to maintain mass conservation, a correction velocity V_c has been introduced to Equation 2.10 as shown in Equation 2.11. The correction velocity is described in Equation 2.12, where N is the number of species [22].

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g Y_i\right) + \nabla \cdot \left(\varepsilon_g (u_g + V_c) \rho_g Y_i\right) = \nabla \cdot \left(\varepsilon_g \rho_g D_i \nabla Y_i\right) + S_{Y_i} + S_{p,Y_i} \quad (2.11)$$

$$V_c = \sum_{i=1}^{N} D_i \nabla Y_i \quad (2.12)$$

2.2.3 Energy conservation

In this study, total energy form has been chose among other forms because of its convenience. The energy conservation equation is given in Equation 2.13. The term α_{eff} in Equation is effective thermal diffusivity.

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g E\right) + \nabla \cdot \left(\varepsilon_g u_g \left(\rho_g E + p\right)\right) = \nabla \cdot \left(\varepsilon_g \alpha_{eff} \nabla h_s\right) + S_h + S_{p,h} + S_{rad} \qquad (2.13)$$

$$E = h_s - \frac{p}{\rho_g} + \frac{u_g^2}{2} \qquad (2.14)$$

2.3 Heat transfer

Modeling of heat transfer can be described under two different categories.

1. Inter-phase heat transfer
2. Radiation modeling

These two different categories are based on modeling of source terms. The convective heat transfer between two phases has been included in source term of inter-phase heat transfer. The convective heat transfer can be due to both temperature gradient and heterogeneous reactions.

Radiation heat transfer is dominant heat transfer method due to high operating temperature of combustor. In this work, P1 model has been included as the governing transport equation for incident intensity, G, with following assumptions.

- Biomass particles (fuel) act as an scattering, emitting and absorbing medium
- The gas phase is optically thin.

2.4 Heterogeneous reactions

2.4.1 Drying

According to Bellais [23], the drying models can be categorized into three different groups.

1. Boiling model.

   Drying occurs at the boiling temperature and that the drying zone is infinitely thin.

3. Equilibrium model.

Water vapor is in equilibrium with the liquid (low-temperature drying).

2. First-order kinetic rate model.

The kinetically controlled rate of drying is calculated by first order Arrhenius expression. Rate constants are evaluated based on experimental data.

Since combustor temperature is usually very high, first two models cannot predict accurate drying rate. Therefore, a drying model based on third category has been chosen with given rate constants as shown in Equation 2.15.

$$r_{dry} = 2.822 \times 10^{-4} exp\left(-10584/T_s\right)|T_s - 475|^7 \tag{2.15}$$

2.4.2 Pyrolysis

$$\frac{dm_{devol}}{dx} = -Aexp\left(-E/RT_p\right)m_{devol} \tag{2.16}$$

Table 1: Volatile gas composition

Component	Mass Fraction
H_2	0.109
CO	0.396
H_2O	0.159
CO_2	0.209
CH_4	0.127

Source: H. Thunman, F. Niklasson, F. Johnsson, and B. Leckner, "Composition of volatile gases and thermochemical properties of wood for modeling of fixed or fluidized beds," Energy and Fuels, vol. 15, no. 6, pp. 1488–1497, 2001. [22]

The second stage of combustion is pyrolysis. Because of not only Euler-Lagrangian computational framework but also actual size (0.05 mm – 3 mm) of biomass particles, first-order kinetic rate model has been chosen as shown in Equation 2.16, where A = 5 x 10^6 1/s and E = 1.2 x 10^8 J/kmol [24] [25]. The volatile gas composition can be found either by available literature (Table 1) [26] [27] or through an experiment, which satisfies Equation 2.17, where $\sum_i \alpha_i = 1$. In here, all pyrolysis hydrocarbons have been lumped to methane while considering bigger hydrocarbon as non-stable products [24] [28].

$$Biomass \rightarrow \alpha_1 CO + \alpha_1 H_2O + \alpha_1 CO_2 + \alpha_1 H_2 + \alpha_1 CH_4 + \alpha_1 char + \alpha_1 ash \quad (2.17)$$

2.4.3 Char combustion

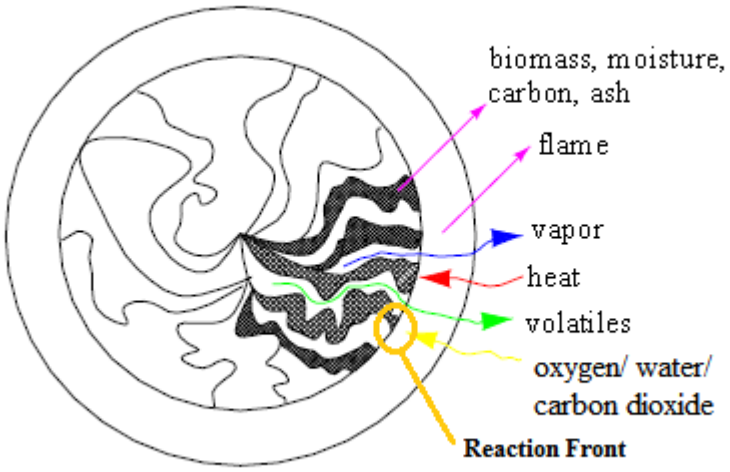

Figure 3: Reaction front of a biomass particle

The third stage is heterogeneous reactions of fixed carbon (char). After drying and devolatilization, fixed carbon and ash are remained in a particle. Since particles are represented by points in the control volume with a diameter, surface reaction of char has been considered instead of volumetric reaction of char (Figure 3). This approach is acceptable according to the [29] [30] [31]. There are three significant heterogeneous reactions; char oxidation, water gasification and carbon dioxide gasification. The char oxidation depends on particle (char) temperature, which is described in Equation 2.18 to 2.20. The equilibrium constant α is called as Boudouard equilibrium constant [32] [6].

$$C + \alpha O_2 \rightarrow 2(1-\alpha)CO + (2\alpha - 1)CO_2 \quad (2.18)$$

$$\alpha = {(2+a)}/{2(1+a)} \quad (2.19)$$

$$a = 2512\exp\left({-6420}/{T_s}\right) \quad (2.20)$$

Water gasification and carbon dioxide gasification are described in Equation 2.21 and Equation 2.22 respectively.

$$C + H_2O \rightarrow 2CO \tag{2.21}$$

$$C + CO_2 \rightarrow 2CO \tag{2.22}$$

The overall heterogeneous reaction rate depends on not only kinetic rate but also mass transfer rate. Therefore overall reaction rate, r_i can be evaluated by parallel resistance of mas transfer rate ($r_{m,i}$) and kinetic rate ($r_{k,i}$) as shown in Equation 2.23 [32].

$$r_i = {r_{k,i} r_{m,i}}/{(r_{k,i} + r_{m,i})} \tag{2.23}$$

The kinetic rate ($r_{k,i}$), where *i* represents gasification or oxidation agent, can be expressed by Equation 2.24. Rate constants are given in Table 2 [33] [34].

$$r_{k,i} = A_c A_i T_s exp\left({-E_i}/{RT_s}\right)\left({M_c}/{M_i}\right)\rho_i \tag{2.24}$$

Table 2: Kinetic data for evaluation of surface reaction rates

Equation	Frequency Factor	Activation Energy (kJ/mol)
2.18	0.652 $ms^{-1}K^{-1}$	90
2.21	3.42 $ms^{-1}K^{-1}$	129.7
2.22	3.42 $ms^{-1}K^{-1}$	129.7

Source: N. Fernando and M. Narayana, "A comprehensive two dimensional Computational Fluid Dynamics model for an updraft biomass gasifier," Renew. Energy, vol. 99, pp. 698–710, 2016. [29]

The term A_c is the effective surface area of char. Since biomass particle is a collection of fixed carbon, ash, moisture and volatile matters, particle diameter is not equal to available surface area for reaction front. Therefore, effective surface area for reaction front has been calculated by Equation 2.25 as a ratio between mass of fixed carbon (m_{char}) and initial mass ($m_{initial}$) [33].

$$A_c = \left({m_{char}}/{m_{initial}}\right)A \tag{2.25}$$

Mass transfer rate ($r_{m,i}$) can be expressed by Equation 2.26, where k_i is mass transfer coefficient [35].

$$r_{m,i} = k_i\left({A}/{6}\right)(\rho_i - \rho_{i,s}) \tag{2.26}$$

This rate equation is based on two assumption. The first assumption is mass diffusion direction. Here, it is assumed that, mass diffusion is significant only in directions that are parallel to the flow. Based on this assumption, specific diffusive surface area has been reduced to 1/6th of specific surface area (A). Second assumption is related to the $\rho_{i,s}$, the density of oxidation or gasification agent at the reaction front. It is assumed that there is no accumulation of oxidation or gasification agent at the reaction front. Therefore, $\rho_{i,s}$ has been considered as zero. The mass transfer coefficient, k_i has been calculated by Equation 2.27, where Sh is the Sherwood number [6].

$$Sh = 2 + 0.6Re^{1/2}Sc^{1/3} \tag{2.27}$$

2.5 Homogeneous reactions

It is true that there are various reaction mechanisms regarding methane and hydrogen combustions. In combustion research area, various research groups are in a race to prove that they present the most accurate mechanism. However, these mechanisms are subjected to various constraints such as temperature range, pressure, premixed conditions etc. Therefore, it is somewhat difficult to generalize any kind of detail mechanisms [36].

With respect to 3-dimensional LES simulation, adding a new specie add a new variable. Normally, it has to be solved for five variables in non-reactive flow; three components of velocity, density and energy term (pressure, temperature, enthalpy, or entropy). If we are going to solve reactive flow with N number of species, then we have to solve additional (N-1) equations. Adding a new chemical reaction only increases the consumption of memory. However, adding a new specie increases not only memory demand but also processor power demand [22] [37].

Therefore, for this study only the following reactions and species are selected based on available literature. Kinetic data are presented in Table 3 [38].

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \tag{2.28}$$

$$CO + \frac{1}{2}O_2 \rightarrow CO_2 \tag{2.29}$$

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \tag{2.30}$$

$$CO + H_2O \rightleftharpoons H_2 + CO_2 \quad (2.31)$$

Table 3: Kinetic data for evaluation of homogeneous reactions

Reaction	Frequency Factor	Activation Energy (kJ/mol)
2.28	5.2×10^{13}	130
2.29	2.32×10^{12}	167
2.30	1.08×10^{13}	125
2.31	12.6	2.78

Source: N. Fernando and M. Narayana, "A comprehensive two dimensional Computational Fluid Dynamics model for an updraft biomass gasifier," Renew. Energy, vol. 99, pp. 698–710, 2016. [29]

Equation 2.31 is reversible reaction, where Table 3 [29] presents the kinetic data to evaluate forward reactions. The equilibrium rate constant for Equation 2.31 is given by Equation 2.32 [39].

$$k_{eq} = 0.029 exp\left(3.40\ x\ 10^7 / RT\right) \quad (2.32)$$

The reactions in gas phase are limited by not only kinetic rate but also turbulent mixing rate. There are few approaches, which include the effect of turbulent mixing rate into overall reaction rate. Among those approaches eddy dissipation model (EDM) has been selected for our CFD model due to its overall acceptability [21]. Based on EDM, turbulent mixing rate is given in Equation 2.33. The *j* represents reactants of reaction *i*. The term ε represents dissipation rate of turbulent kinetic energy since this equation is based on k-ε turbulence model. However, ε can be replaced by alternative dissipation rate term of turbulent kinetic energy according to the turbulence modeling [21] [22].

$$r_{t,i} = 4\rho_g\left(\varepsilon/k\right) min\left(Y_j / M_j\right) \quad (2.33)$$

The overall reaction rate can be found by taking minimum between turbulent mixing rate ($r_{t,i}$) and kinetic reaction rate ($r_{k,i}$).

2.6 Turbulence modeling

Solving turbulence is the most difficult part in computational fluid dynamics. There are three different modeling approaches namely, Reynolds-averaged Navier–Stokes

(RANS), Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). Most studies have been carried out with RANS modeling because of less computational cost and desirability of averaged results. DNS modeling is not possible today with this kind of combustor geometry due to higher demand of computational power [40] [16]. As shown in Figure 4, LES modeling resolves eddies which are larger than LES length scale.

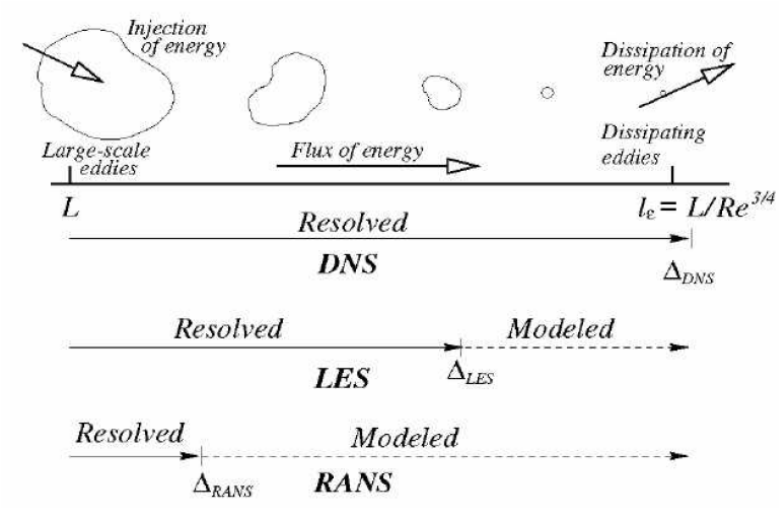

Figure 4: Difference between turbulence modeling approaches

Source: J. Sodja, "Turbulence models in CFD," University of Ljubljana, 2007. [41]

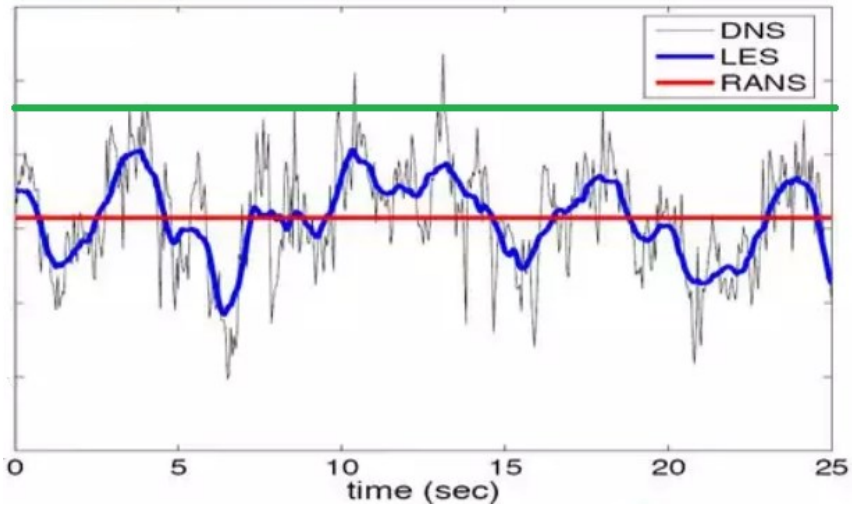

Figure 5: Variation of a variable with respect to turbulence modeling

Source: T. Poinsot and D. Veynante, Theoretical and Numerical Combustion, 2nd ed. PA, USA: Edwards Inc., 2005 [16].

However, it is more appropriate to select LES modeling over RANS modeling despite that it requires more computation power. The reason is not only overall reaction rate heavily depends on fluid phase turbulence but also safety of combustor. Figure 5 shows variation of a variable in fluid phase with different turbulence modeling over time (DNS can be considered as actual value). If this is a temperature of a near wall location, then RANS modeling only give average value, which is always below the maximum allowable temperature (green line) of construction material. However, actual value is higher than upper limit. Therefore, it is not possible to sustain with RANS modeling.

Although LES modeling captures fluctuation of variables, results of LES CFD simulation are, mesh dependent at walls. Therefore, Detached Eddy Simulation (DES) modeling has been used to resolve turbulence in near wall region. In DES, the solver generally uses LES to resolve turbulence. However, in near wall region, it uses LES only if mesh is fine enough. Otherwise, it resolve turbulence by RANS modeling (here k-omega model) [42] [43] [44].

The k-omega SST DES model has been used to resolve turbulence. The term SST stands for Shear Stress Transport, where can be used as Low-Re model especially in near wall region. The original model has been modified to capture effect of discrete particle phase [45] [46] [44].

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g k\right) + \nabla \cdot \left(\varepsilon_g u_g \rho_g k\right) = \nabla \cdot \left(\varepsilon_g \left(\mu_g + {}^{\mu_t}/_{\sigma_k}\right)\nabla k\right) + P_k + \beta_k k \omega F_{DES} \qquad (2.24)$$

$$\frac{\partial}{\partial t}\left(\varepsilon_g \rho_g \omega\right) + \nabla \cdot \left(\varepsilon_g u_g \rho_g \omega\right) = \nabla \cdot \left(\varepsilon_g \left(\mu_g + {}^{\mu_t}/_{\sigma_{\omega 1}}\right)\nabla \omega\right) + P_\omega + \beta_\omega \omega^2 + 2(1-F)\sigma_{\omega 2}\frac{1}{\omega}\nabla \omega \nabla k \qquad (2.25)$$

3. NUMERICAL SOLUTION TECHNIQUE

As described above governing equations are nonlinear. Therefore, numerical solution techniques have to be employed to solve the system of equations. Although there are some standard algorithms, we have to modify them with respect to nature and the complexity of the problem.

This chapter provides descriptions of numerical techniques that have been used to obtain solution for fluidized bed model equations and OpenFOAM C++ toolbox [13], which provides finite volume discretization and many more. Especially, a comprehensive discussion is carried out on especial tricks that have been found helpful in obtaining the accurate results with lesser computer burden. The custom solution algorithms are outlined at the end of the chapter.

3.1 Introduction to finite volume method

A set of partial differential equations can be numerically solved by various techniques commonly called by discretization methods. Among those various discretization methods, finite volume (FVM) method is more commonly used in the field of CFD because of involvement of conservation of transport equations. A generalized governing equation for variable ϕ can be presented as in Equation [21].

The space, which is bounded by known boundary values, is divided into small sections, which are called cells. These cells individually act as control volume. Then the Gauss Divergence theorem is applied over control volumes to make the matrix form of the governing equations. These matrixes are solved for the conservation of the variable ϕ (vector or scalar) under satisfaction of boundary conditions for a given time difference (Δt). The accuracy of a solution, which is obtain for a given time difference is controlled by a convergence criterion.

3.1.1 Discretization schemes

There are few different approaches to interpolate cell center value into cell faces (finite difference methods) such as centered scheme, backward scheme etc. All these schemes have two characteristics in more or less.

1. Dissipation

   Wave is attenuated by the scheme.

2. Dispersion

   Speed of wave is affected by the scheme.

In a numerical simulation, especially with LES turbulence modeling, dissipation schemes should not be used at all. It is difficult to avoid dispersion of waves. Therefore, we select centered scheme over other schemes because centered schemes are not dissipative. However, since we used second order schemes, we could not avoid dissipation completely.

3.2 OpenFOAM solver

Obtaining numerical solutions cannot be done by manual calculations. Therefore, OpenFOAM C++ CFD toolbox is chosen as an automated CFD tool. It provides the communication interface between user and the High Performance Computer. It is well established Linux based CFD toolbox written by C++ computer language. Although it provides the basic solvers and other related functionalities, a custom solver has to be made to address complexity of the problem.

As shown in Figure 6, entire software package is divided into several C++ classes. These classes are allocated for different functionalities, as an example Finite Volume class and its sub-classes handle the discretization.

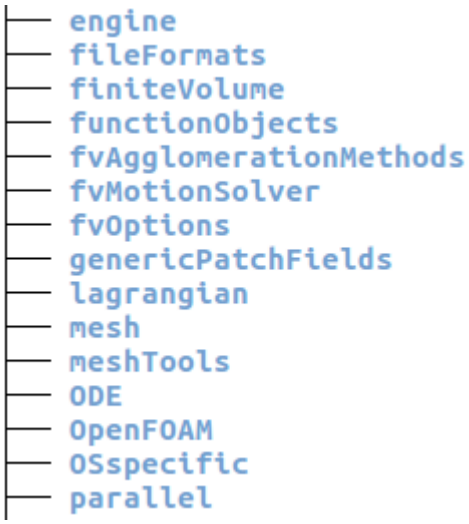

Figure 6: Fraction of OpenFOAM directory tree

3.3 OpenFOAM case

As mentioned earlier, OpenFOAM uses Linux file management system. Therefore, every case uses unique file handling mechanism as shown in Figure 7. The '0' folder and its files contain initial and boundary conditions of dependent variables. The 'system' folder and its files handle the control functions such as convergence criteria, solver parameters etc. The other constant properties, which are not going to change throughout the run time, are located in the 'constant' folder. The 'processor' folders are sub folders, which handle decomposed domains. (In Figure 7, it only shows 7 processor fodders. However, actual case was decomposed into 23 processors.)

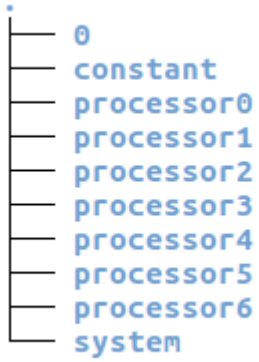

Figure 7: Case directory tree

At every discretized time step, solver solves until convergence criterion is satisfied. However, results are not written to the storage disk unless instructions are given to write at that time. Since, this particular case is dynamic case, it is mandatory to write out in reasonable minor time steps to analyze the dynamic over time.

3.3.1 Fluidized bed biomass combustor

The simulation geometry is rectangular box of dimensions 1.5 m (height), 0.23 m (width) and 0.0015 m (thickness). Since this study has been restricted to quasi-3D, thickness of the simulation geometry is limited to diameter of one particle. However, both front and back faces have been treated as symmetrical patches in order to obtain results that are more realistic. The boundary conditions are listed in Table 4.

Table 4: Boundary conditions

Boundary	Velocity	Pressure	Temperature
Inlet patch	Fixed velocity	Zero gradient	Fixed temperature
Outlet patch	Zero gradient	Fixed pressure	Zero gradient
Wall	No slip condition	Zero gradient	Fixed gradient

4. MODEL VALIDATION

The validity of CFD simulation results are depended on the amount of trust earned by the CFD solver. Therefore, we compared simulation results from our CFD solver with experimental results not only from our own experimental setup but also available in literature.

4.1. Validation Case 1

The CFD solver was validated by comparing two temperature profiles, which obtained from a CFD simulation and a trail carried out by the prototype of sawdust combustor as shown in Figure 8. This continuous combustor was made and tested by the National Engineering Research and Development Centre (NERD) of Sri Lanka. The geometry of prototype is shown in Figure 9. This combustor differs from other types of combustors due to its unique design shape, which include two chambers. It is designed to provide a heat output in the range of 250 kW to 750 kW with the paddy husk consumption of 60 kg/h to 180 kg/h. The temperature of flue gas from the burner is around 1000 °C. Thermal efficiency of burner was found to be 80% [47]. Two of suspension combustors have been installed at the Rice Processing Research and Development Centre (RPRDC) and Ariya rice mill in Anuradhapura.

First, combustor was preheated up to 600 °C, by firing firewood. Approximately 10 kg of firewood was burnt within a period of around ½ hour to heat up until 600 °C. Then feeding rate of sawdust was regularly increased with increasing air supply. The air to fuel ratio was manually controlled. The particle size distribution and chemical properties of biomass are described in Table 5 & 6. The temperature variation at T2 (Figure 8) throughout the trial, is shown in Figure 10. A comparison between simulated and experimental temperature profiles at T1, T2 and T3 are presented in Figure 11. Based on Figure 11, we can conclude that presented model has predicted experiment with reasonable accuracy.

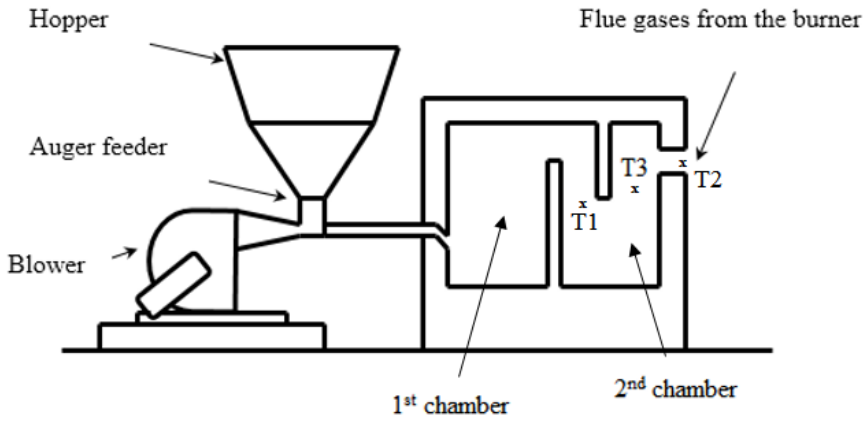

Figure 8: Diagram of combustor prototype

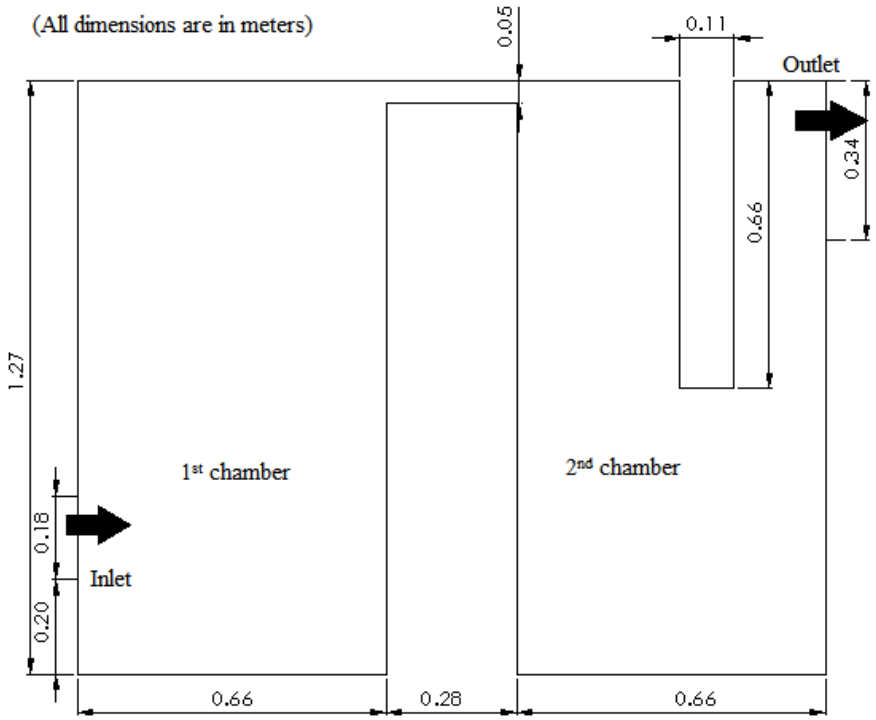

Figure 9: Combustor geometry

Table 5: Particle size distribution

Size (mm)	> 2.0	2.0 - 0.63	0.63 - 0.2	0.2 - 0.063
Weight Fraction (%)	1	28	53	18

Table 6: Proximate analysis

Composition	Moisture	Volatile matter	Fixed carbon	Ash
Weight Fraction (%)	20	65.73	13.48	0.79

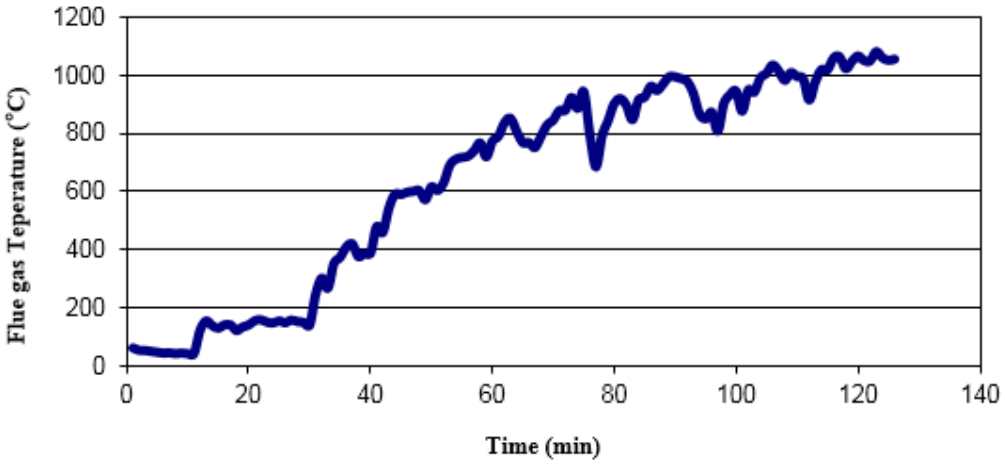

Figure 10: Flue gas temperature

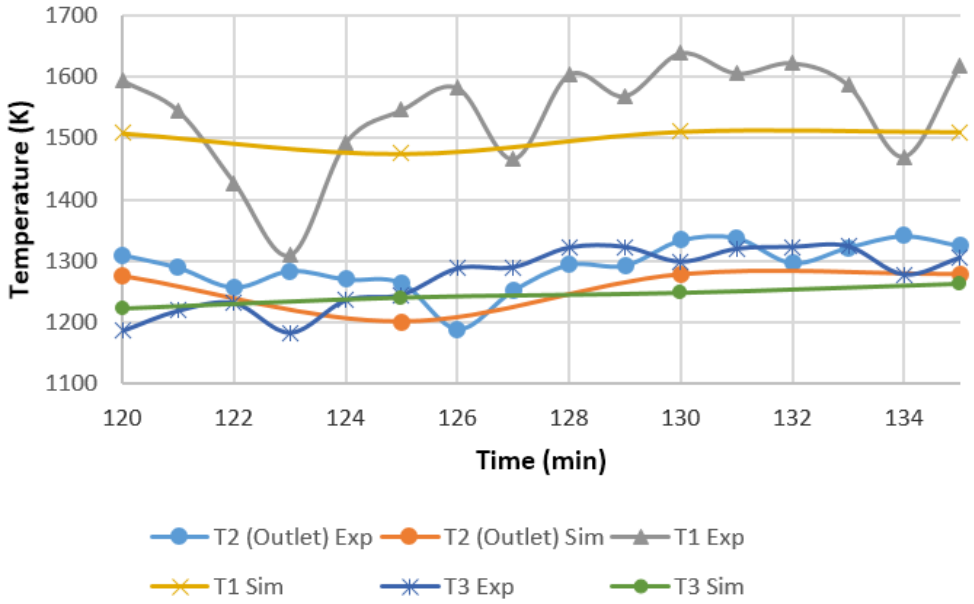

Figure 11: Steady state temperature profiles at T1, T2 & T3

4.2. Validation Case 2

Since both discrete phase and continuous phase have to be considered when modeling, we have to use Eulerian–Lagrangian approach. As described in Chapter 2, MP-PIC (Multiphase Particle-In-Cell) method has been chosen over DPM (Discrete Particle Method) because of less computer burden. Although it is less expensive, some of discrete properties may be lost. However, following results show a comparison between fluidization behavior of a particle bed from MPPIC simulation and of from published DPM simulation under same operating conditions [48], [49], [50].

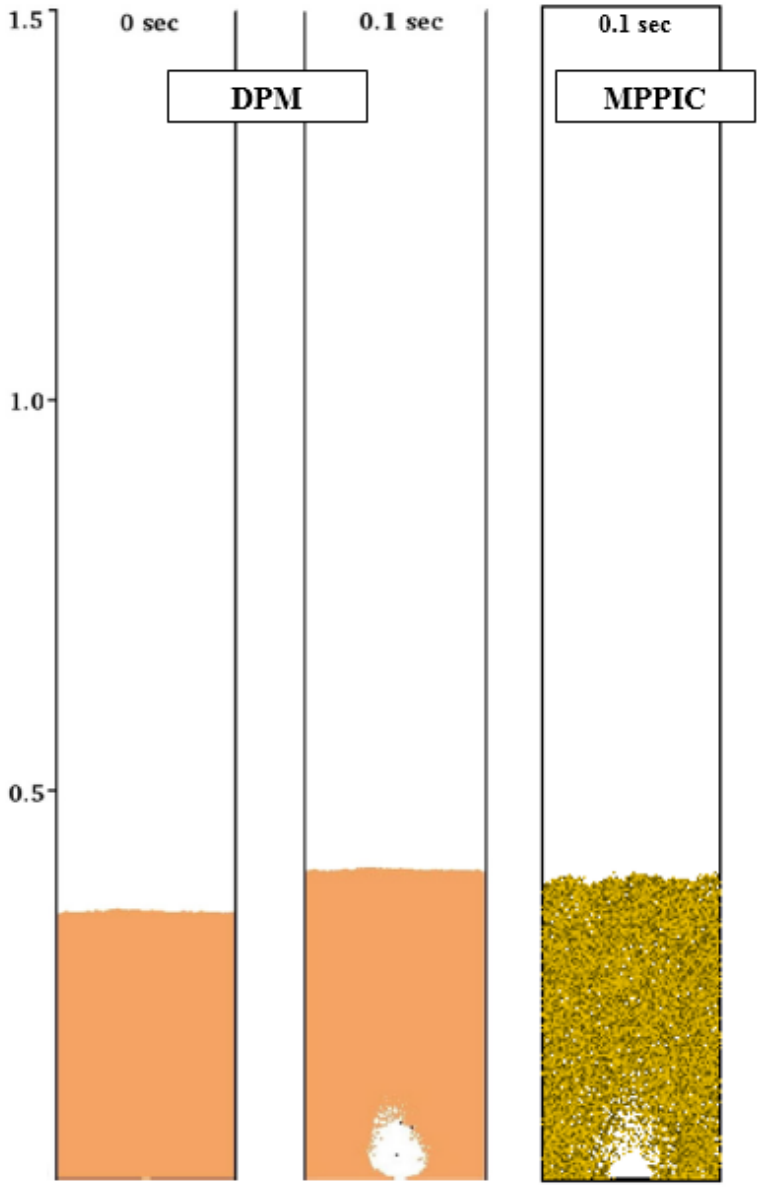

Figure 12: Particle flow pattern (A)

The fluidization pattern can be analyzed by investigating the formation and propagation of air bubble through initial packed bed. Figure 12 shows initial packed bed with simulated flow pattern after 0.1 second. It presents results from both DPM and MPPIC methods side by side. The formation of bubble can be seen clearly in DPM simulation. However, MPPIC also predicts the bubble in somewhat dissipative way. The increments of bed height and bed symmetricity are almost same in both simulations.

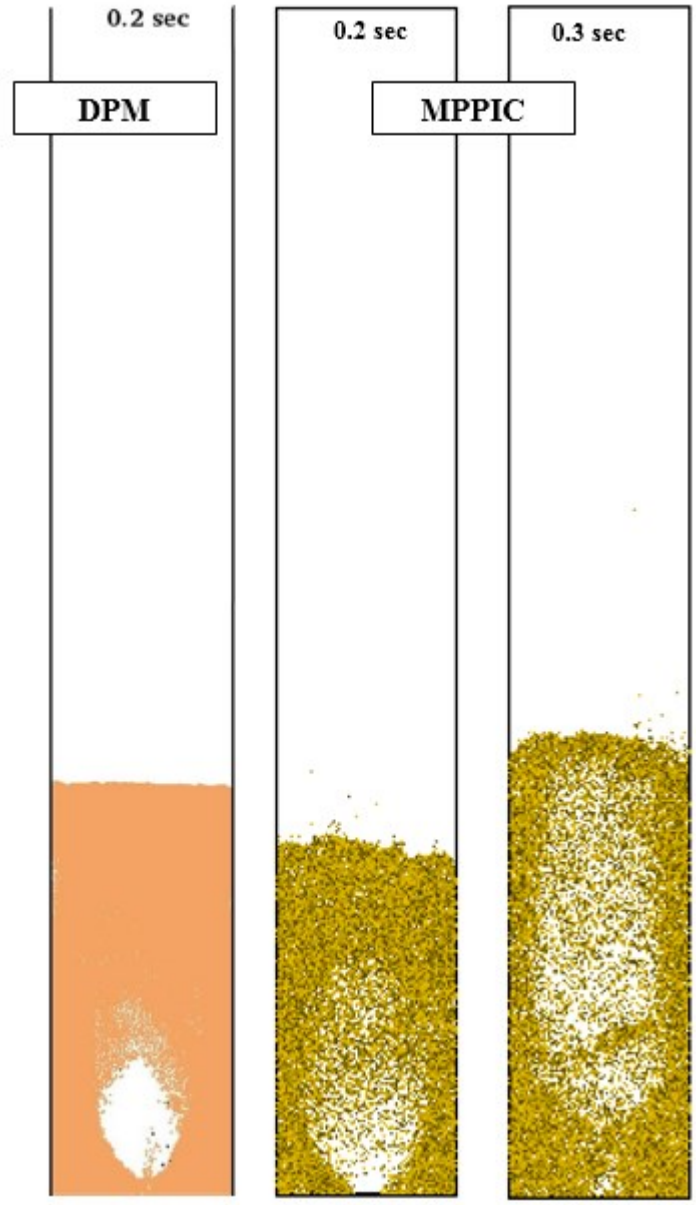

Figure 13: Particle flow pattern (B)

Figure 13 shows flow pattern after 0.2 second and 0.3 second. After 0.2 second, DPM simulation predicts developed bubble that detached from inlet. However, MPPIC simulation predicts developed dissipative bubble that attached to inlet. The prediction of bed height increment is less in MPPIC simulation than that of DPM simulation. The reason for this kind of behavior in MPPIC simulation is its stress model. DPM simulation calculates individual particle collisions, which give results that are more accurate.

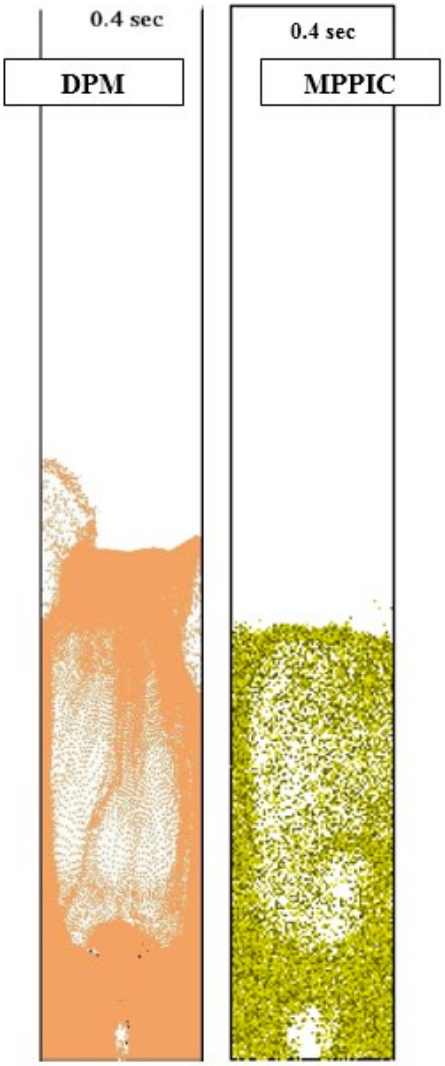

Figure 14: Particle flow pattern (C)

In Figure 14, both simulations have predicted collapse of bubble, which leads to sluggish structure. Although fluidized height has not been perfectly predicted, structural changes have been predicted accurately by MPPIC simulation.

4.3. Validation Case 3

This validation case is based on published experimental data [51] and simulation data [48] on steam gasification. However, same CFD solver can be used to simulate both gasification and combustion by alternating reaction schemes. Since oxygen is absent in gas phase char oxidation (Equation 2.18) has no impact at all. Fluid phase homogeneous reactions have been restricted to steam reforming and water gas shifting.

Figure 15 shows the improvement of steady state results with improved modeling, because of more accurate discrete particle modeling and LES turbulence modeling. The purple columns represent simulated results with LES turbulence modeling, which are almost same as experimental volume fraction of product gas.

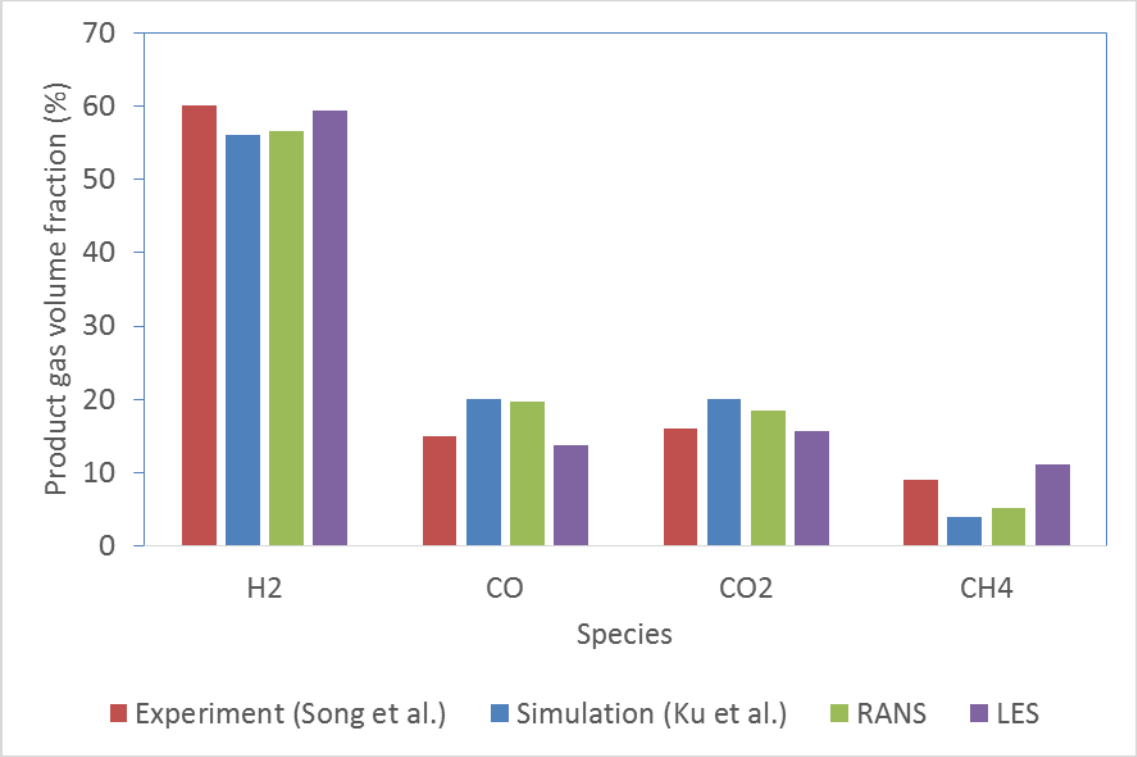

Figure 85: Product gas composition at exhaust

5. RESULTS & DISCUSSION

5.1. Prediction of system parameters

The CFD solver can be used to predict the system parameters of a fluidized bed combustor operated under known conditions. This was tested using a laboratory scale unit, which was built and operated in 24-hours laboratory in the department premises. A sketch of that unit is shown in Figure 17.

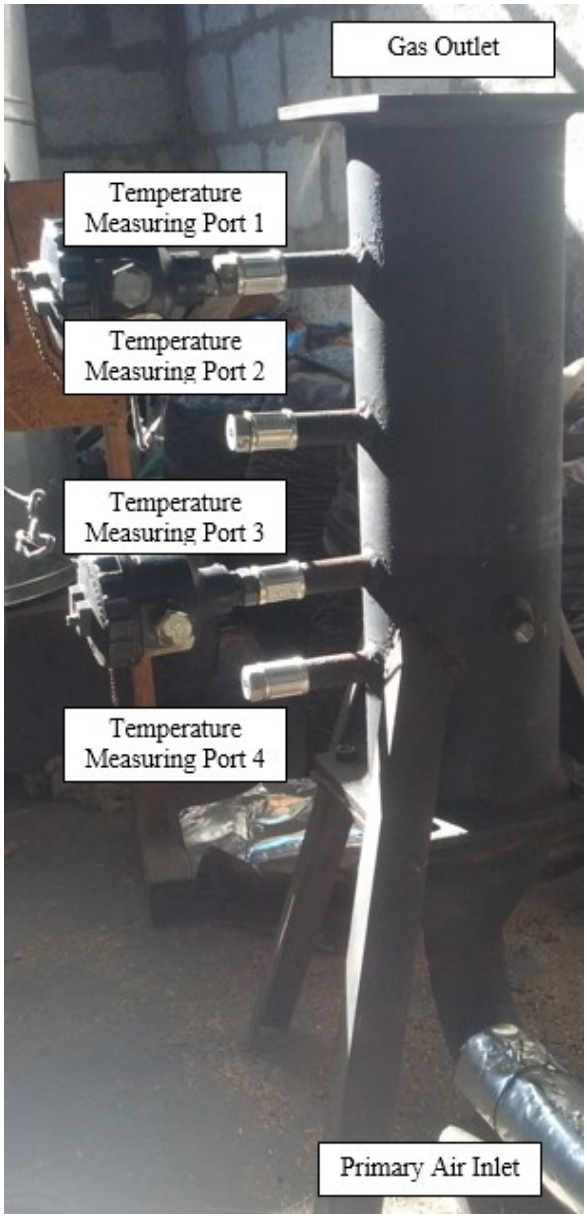

Figure 16: Experimental setup

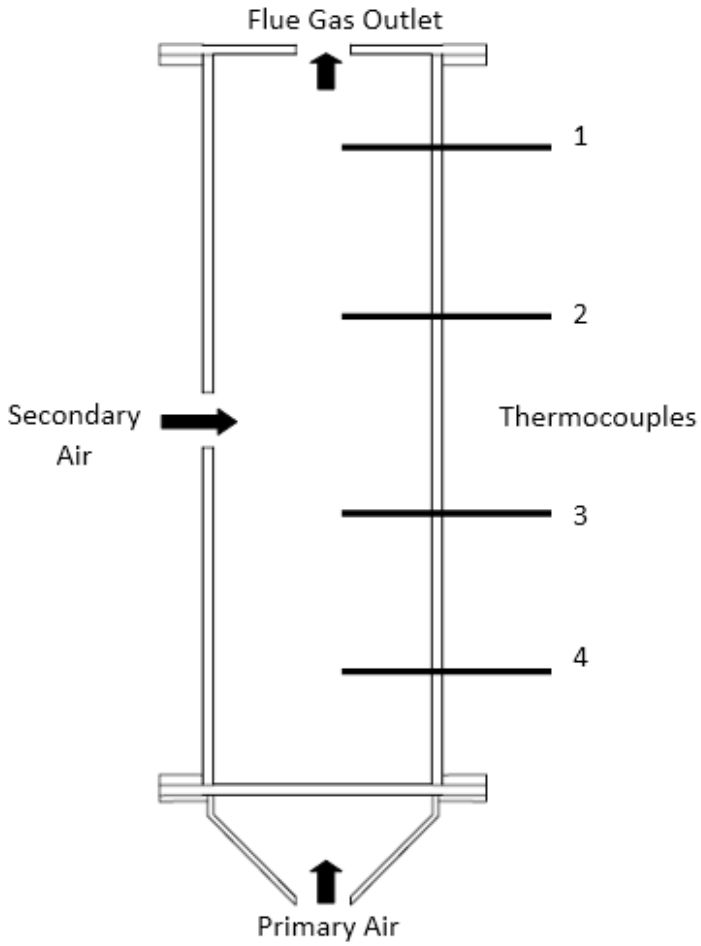

Figure 17: Sketch of experimental setup

The combustion geometry was a cylindrical unit, which had inside diameter of 0.128 m, wall thickness of 6.5 mm and height of 0.48 m (5 NPS/ SCH 40 steel tube). A conical air distributor, which had an angle of 60°, was used to distribute primary air followed by a stainless steel mesh, which was used as a holding plate for biomass particles. The secondary air inlet was 0.25 m below the flue gas. Four temperature measurements were made as shown in Figure 16 and Figure 17. The first measuring port was 0.1 m below the flue gas outlet while fourth one was 0.11 m above the distributed plate. The gaps between first three measuring ports were 0.1 m each.

Initially, combustor filled with a packed biomass (sawdust) bed of 162.1 g in weight and 0.11 m in height, which is composed of nearly spherical particles. The particle size distribution and chemical properties of biomass are described in Table 5 and Table 6. It is assumed that, sawdust bed is initially a packed bed. At the air inlet (bottom of the

combustor), 1 ms^{-1} of air flow rates was maintained. Biomass bed was manually ignited from the top at the beginning of experiment.

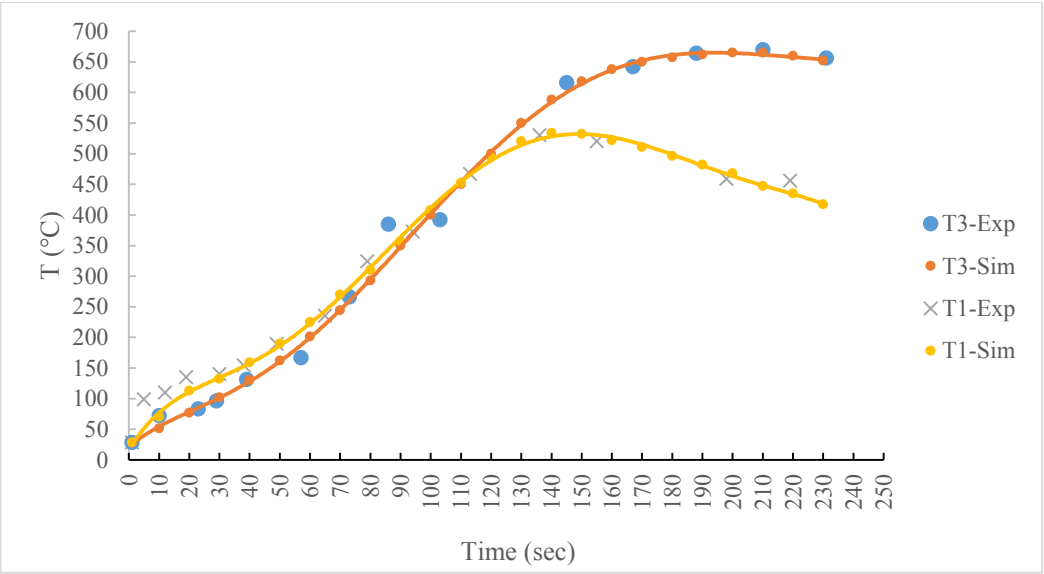

Figure 18: Combustor temperature at T1 & T3 with time

As shown in Figure 18, predicted temperature profiles almost perfectly match each with measured temperature profiles. However, in this case study wall heat loss was adjusted to obtain above result because measuring heat loss was not possible with available facilities.

The important outcome from this case study is if we know correct boundary conditions then we can predict accurately. In other word, we can predict unknown system parameters by using proposed CFD solver.

5.2. Optimization

This study intends to develop a three dimensional CFD model to analyze fluidized bed combustion of biomass, which can demonstrate the turbulence in the freeboard in order to find optimum process parameters (excess air ratio and primary air to secondary air ratio). Since the research was focused on effect of turbulence of the freeboard, LES (Large - Eddy Simulation) turbulence modeling was included into CFD code. As shown in Figure 15 steady state outlet gas composition in validation case 4 is more accurate when using LES turbulence modeling. This improvement can be justified by results obtained from biomass combustor case as shown in Figure 19. Even though

these shown eddies are not related to validation case 4, the reason for higher accuracy is better mixing due to eddies. This kind of prediction cannot be expected from RANS turbulence modeling.

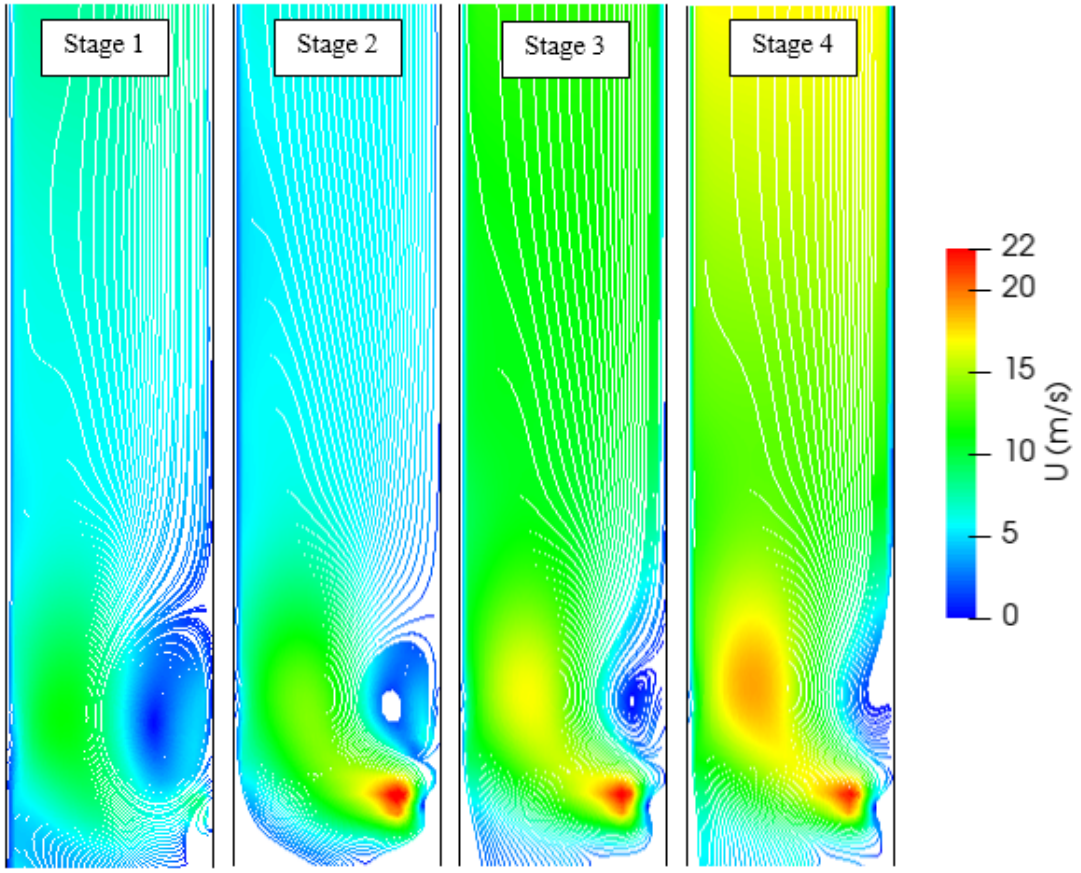

Figure 9: Eddies near secondary air inlet

In this study, two process parameters have been optimized namely, excess air ratio and primary air to secondary air ratio. As shown in simulated results (Figure 20), optimum excess air ratio can be found as 10%, which gives freeboard temperature of 1400 K and CO level of 1ppm. If we increase the excess air ratio, CO level becomes stable around 50 ppm and freeboard temperature reduces continuously. This behavior is due to lack of residence time and can be observed clearly in simulated results (Figure 21).

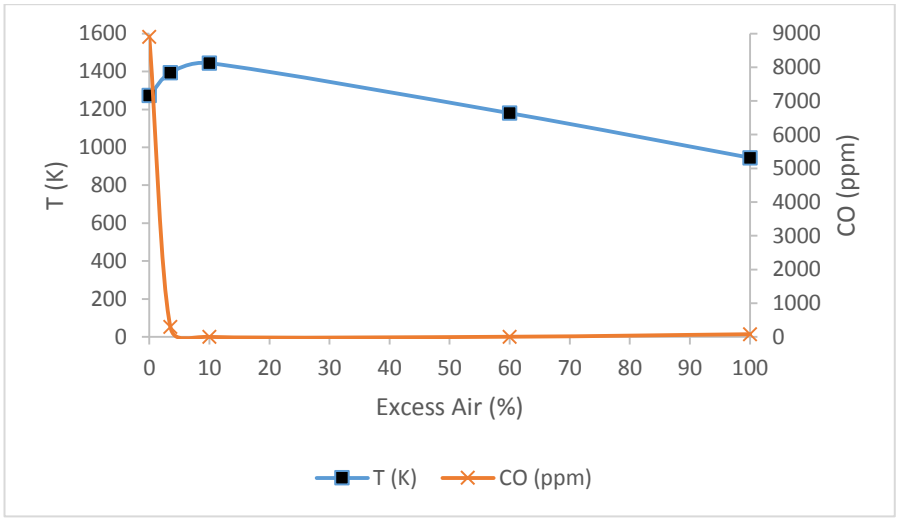

Figure 10: Freeboard temperature and exhaust CO level for different excess air ratios

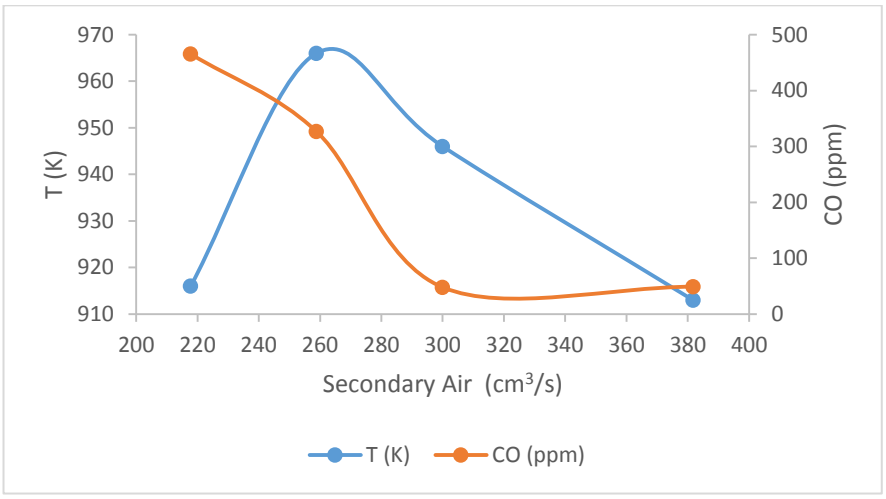

Figure 11: Freeboard temperature and exhaust CO level for different secondary air flow rates

After a threshold value (300 cm^3/s), exhaust CO level remains around steady value (50 ppm) due to lack of residence time. However, after another threshold value (400 cm^3/s), CO level increases beyond this steady value because residence time becomes insufficient to consume remaining CO in the freeboard.

Figure 22 and Figure 24 show temperature profile and CO profile for different excess air ratios. Figure 21 is clearly presented lower freeboard temperate at high excess air ratio (100%). As shown in Figure 24, at high excess air ratio (100%), there is unburnt CO in freeboard due to lack of residence time.

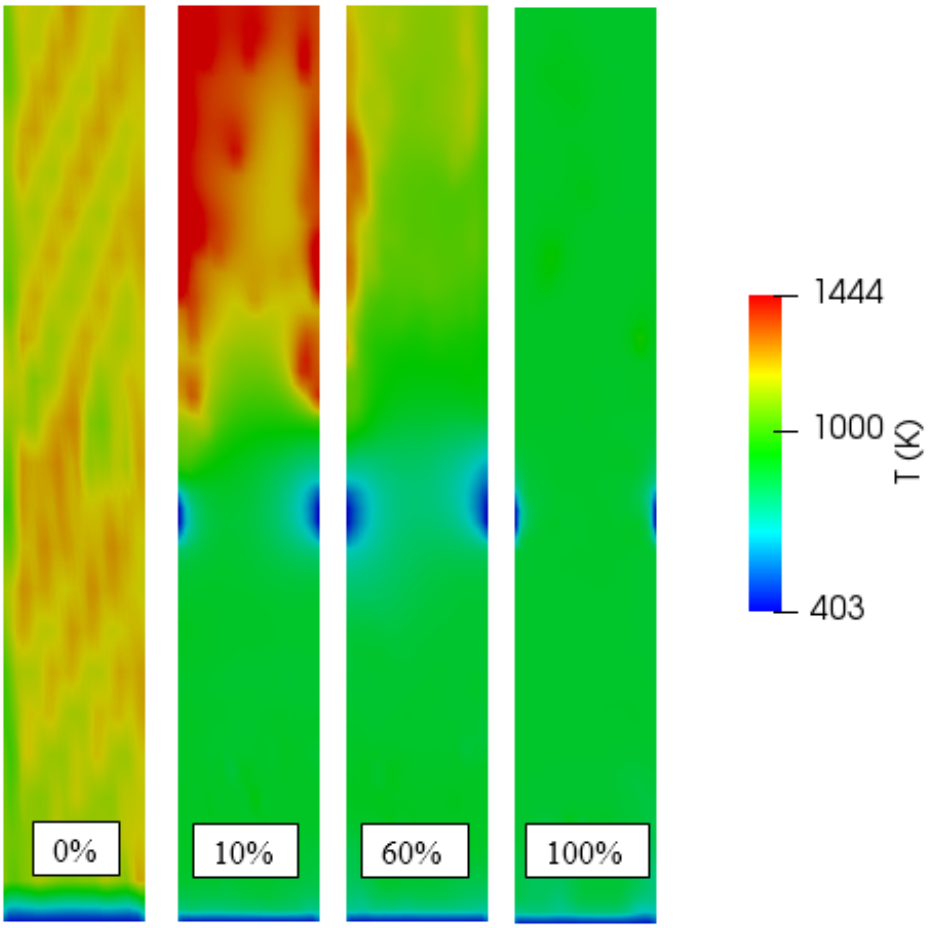

Figure 12: Temperature profiles for different excess air ratios

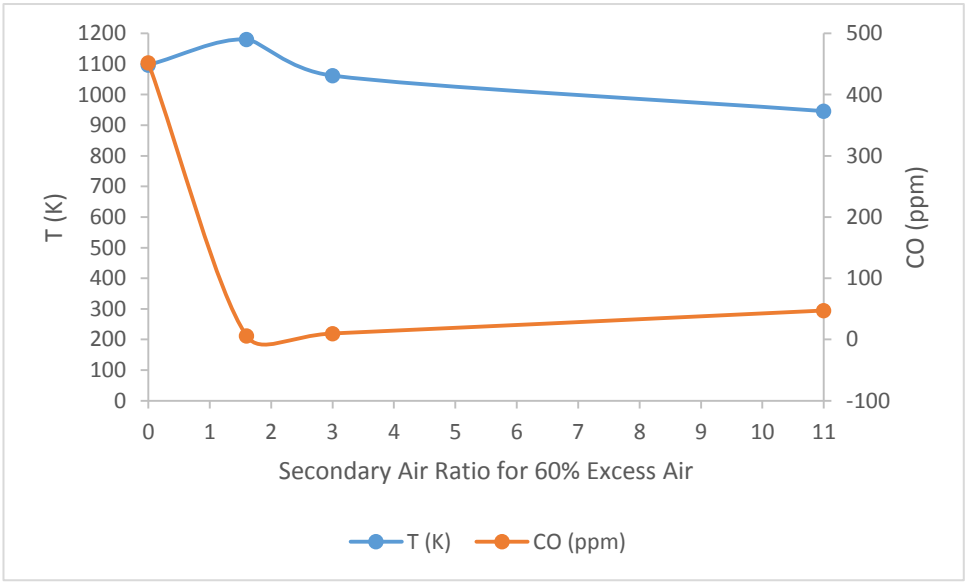

Figure 23: Freeboard temperature and exhaust CO level for different secondary air ratios (60% excess air)

The second process parameter is ratio between secondary air and primary air. As shown Figure 23, freeboard temperature decreases when the secondary air ratio increases. It is due to the exhaust loss. The same Figure shows high exhaust CO level at low secondary air ratio. Low secondary air ratio for same excess air (60%) means higher primary airflow rate. Therefore, this high CO level is caused by higher char conversion in pyrolysis region.

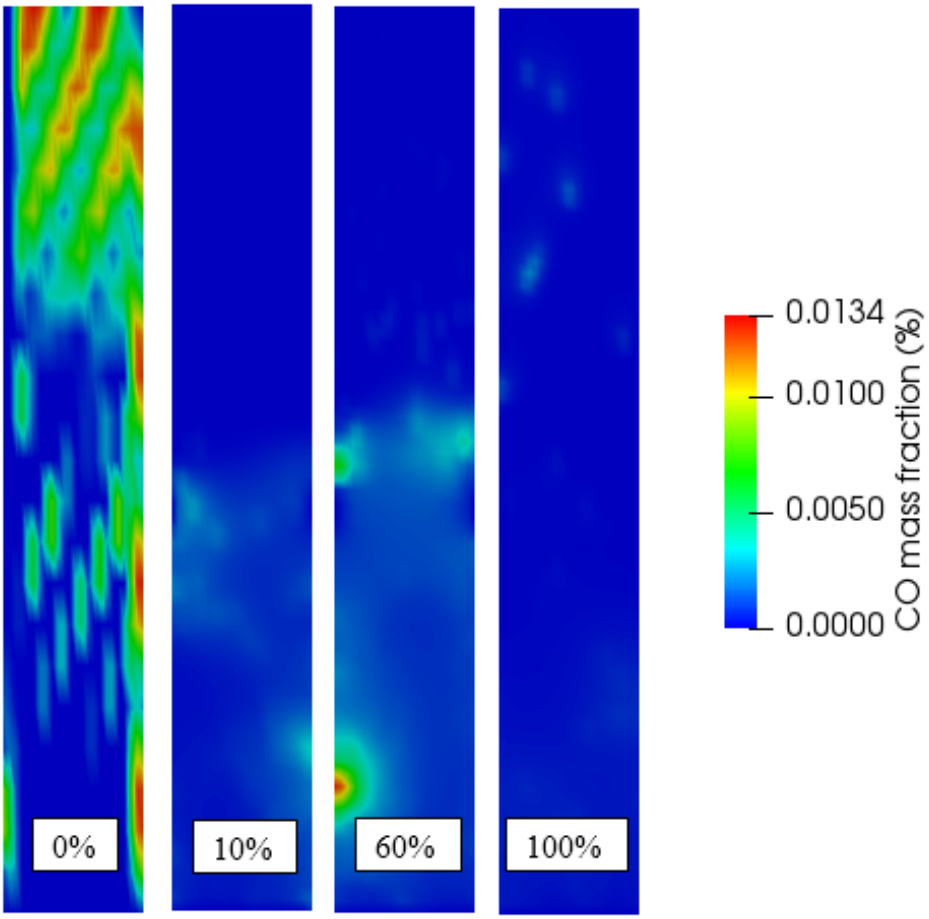

Figure 24: CO profile for different excess air ratios

5.1 Suspension combustion – case study

Figure 13: Suspension combustor prototype

The main objectives of this case study were analyzing the performance of above combustor (Figure 8 & Figure 25) and improving the overall thermal efficiency. A fraction of work done under this case study has been presented previously in Section 4.1 as Validation Case 1. Since CFD analysis result spatial distribution of variables, geometrical optimization is much efficient than conventional experimental methods.

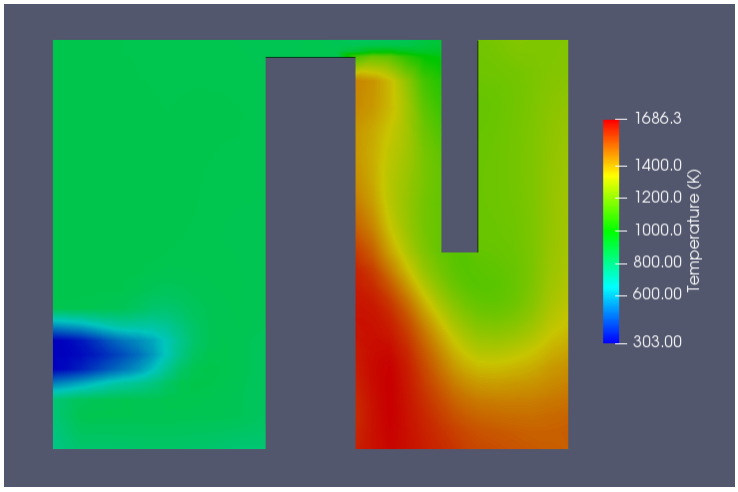

Figure 14: Steady state temperature profile

Figure 26 shows steady state temperature profile that completely matches with Figure 11. Figure 27 shows turbulent kinetic energy profile at steady state. When considering

Figure 26 and Figure 27, we have obtained higher temperature where turbulent kinetic energy at its maximum value.

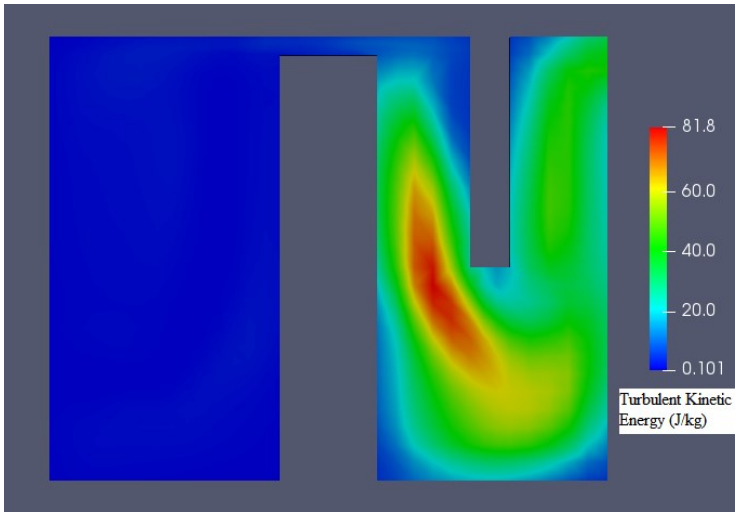

Figure 15: Steady state turbulent kinetic energy profile

Figure 28 shows particle size distribution throughout combustor while Figure 29 presents residence time distribution. When we know the size of particles at exhaust, we can implement better treatment for particulate matters. According to the Figure 28, size of particles at outlet is significantly low with respect to inlet. This difference implies higher efficiency of combustion process.

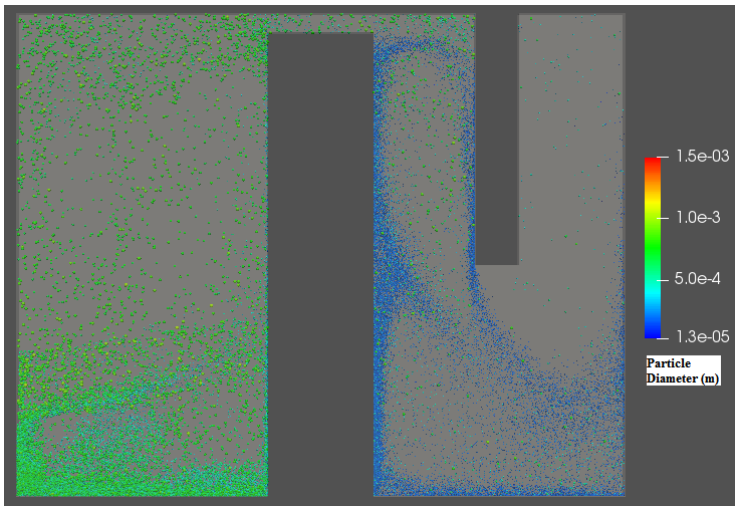

Figure 16: Steady state particle size distribution

Figure 30 show mass fraction of CO in combustor. In this case study, we did not calculate NO_x formation. Therefore, CO is the only pollutant, which should be

controlled while getting maximum thermal conversion. As shown in Figure 30, the mass fraction of CO at exhaust is null, which indicates clean exhaust out. However, there is some unburnt methane at outlet in negligible amount. After series of simulations, optimum airflow rate was found as 5.5 ms^{-1} for a 0.00171 kg/s of fuel feeding rate (65% amount of excess air) [52].

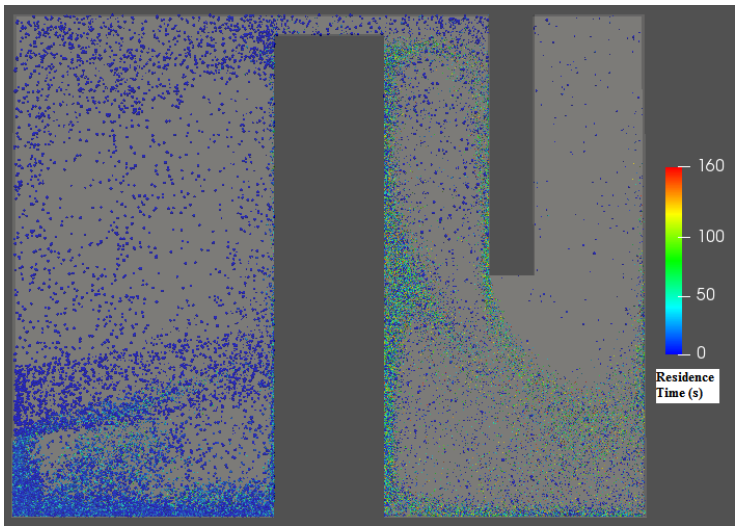

Figure 17: Steady state residence time distribution

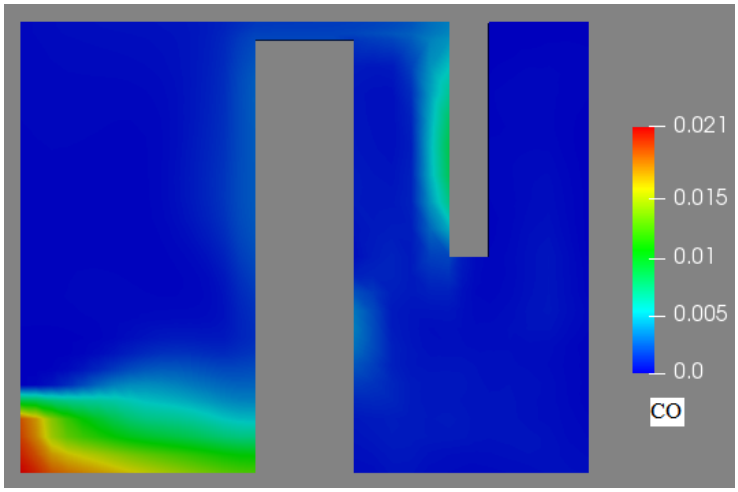

Figure 18: Steady state CO profile

6. CONCLUSION & FUTHER WORKS

6.1 Conclusion

The Eulerian-Lagrangian CFD modeling of biomass combustion has been succeeded with OpenFOAM. The present model can predict dynamics of freeboard with both biomass combustion and homogeneous reactions. The turbulence modeling has significant impact on predicted results since turbulence mixing change overall reaction rate completely. There is huge improvement in both steady state results and dynamic results when LES turbulence modeling has been implemented instead of RANS turbulence modeling. We have used Detached Eddy Simulation, which predict better results at near wall regions without less mesh resolution.

The performance of fluidized bed biomass combustor depends on the turbulence of freeboard: the amount of secondary air. The optimum amount of secondary air ratio is 1.6%, which gives higher freeboard temperature while maintaining minimum emission. The optimum excess air ratio is 10% for 0.1125 kg/h.

In the case of suspension combustion, maximum combustor temperature has been obtained for optimum airflow, which was 5.5 ms^{-1} of inlet air velocity for a 0.00171 kg/s of fuel feeding rate (65% amount of excess air).

6.2 Further works

Modeling of reactive flow (combustion) with LES turbulence has many difficult challenges. Therefore, this CFD model can be improved by implementing higher order numerical schemes (at least 4^{th} order) than second order. A complete emission profile can be obtained by including a more complete reaction scheme.

References

[1] S. B. Kausley and A. B. Pandit, “Modelling of solid fuel stoves,” *Fuel*, no. 89, pp. 782–791, 2010.

[2] M. Miltner, A. Makaruk, M. Harasek, and A. Friedl, “Computational fluid dynamic simulation of a solid biomass combustor: Modelling approaches,” *Clean Technol. Environ. Policy*, vol. 10, no. 2, pp. 165–174, 2008.

[3] J. Chartier, P. Guernion, and I. Milo, “CFD modelling of municipal solid waste incineration,” in *Progress in Computational Fluid Dynamics 7*, 2007, pp. 19–24.

[4] K. Jang, W. Han, and K. Y. Huh, “Simulation of moving-bed and fluidized-bed reactors by DPM and MPPIC in OpenFOAM,” in *11th OpenFOAM Workshop*, 2016.

[5] J. F. Davidson and D. Harrison, *Fluidized particles*. Cambridge: Cambridge University Press, 1963.

[6] S. N. Oka, *Fluidized Bed Combustion*, 1st ed. New York: Marcel Dekker, 2004.

[7] C. Y. Wen and Y. H. Yu, “Mechanics of fluidization,” in *Chem. Eng. Prog. Symp. Ser.*, 1966, pp. 100–111.

[8] C. G. Philippsen, A. C. F. Vilela, and L. D. Zen, “Fluidized bed modeling applied to the analysis of processes: Review and state of the art,” *J. Mater. Res. Technol.*, vol. 4, no. 2, pp. 208–216, 2015.

[9] W. C. Yang, *Handbook of fluidization and fluid-particlesystems*. New York: Taylor & Francis, 2003.

[10] D. Kunii and O. Levenspiel, *Fluidization engineering*. New York: Jonh Wiley, 1969.

[11] C. E. J. van Lare, “Mass transfer in gasfluidized beds: scaling, modeling and particle size influence,” Technische Universiteit Eindhoven, 1991.

[12] M. Mnndo, L. Rosendahl, C. Yin, and H. Sorensen, “Pulverized straw combustion in a low-NOx multifuel burner: Modeling the transition from coal to straw,” *Fuel*, no. 89, pp. 3051–3062, 2010.

[13] The OpenFOAM Foundation, “OpenFOAM dev,” 2018. [Online]. Available: https://cpp.openfoam.org/dev/. [Accessed: 11-Dec-2018].

[14] J. K. A. T. Rajika and M. Narayana, “Modelling and simulation of wood chip combustion in a hot air generator system,” *Springerplus*, vol. 5, no. 1, p. 1166, 2016.

[15] A. Berlemont, P. Achim, and Z. Chang, “Lagrangian approaches for particle collisions: The colliding particle velocity correlation in the multiple particles tracking method and in the stochastic approach,” *Phys. Fluids*, vol. 13, no. 10, pp. 2946–2956, 2001.

[16] X. Wang, B. Jin, and W. Zhong, “Three-dimensional simulation of fluidized bed coal gasification,” *Chem. Eng. Process. Process Intensif.*, vol. 48, no. 2, pp. 695–705, 2009.

[17] S. Elghobashi, “On predicting particle-laden turbulent flows,” *Appl. Sci. Res.*, vol. 52, no. 4, pp. 309–329, 1994.

[18] M. J. Anderews and P. J. O’Rourke, “The multiphase particle-in-cell (MP-PIC) method for dense particulate flows,” *Int. J. Multiph. Flow*, vol. 22, no. 2, pp. 379–402, 1996.

[19] S. Benzarti, H. Mhiri, and H. Bournot, “Drag models for Simulation Gas-Solid Flow in the Bubbling Fluidized Bed of FCC Particles,” *Waset.Org*, vol. 61, no. 1, pp. 1138–1143, 2012.

[20] S. Ergun, “Fluid flow through packed columns,” *Chem. Eng. Prog.*, vol. 48. pp. 89–94, 1952.

[21] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics*, 2nd ed. Essex: Prentice Hall, 2007.

[22] T. Poinsot and D. Veynante, *Theoretical and Numerical Combustion*, 2nd ed.

PA,USA: Edwards Inc, 2005.

[23] M. Bellais, “Modelling of the pyrolysis of large wood particles,” KTH - Royal Institute of Technology, 2007.

[24] S. Gerber, M. Oevermann, and F. Behrendt, “An Euler-Lagrange modeling approach for the simulation of wood gasification in fluidized beds,” in *4th European Combustion Meeting*, 2009, pp. 1–6.

[25] N. Prakash and T. Karunanithi, “Kinetic modeling in biomass pyrolysis - a review,” *Appl. Sci. Res.*, vol. 4, no. 12, pp. 1627–1636, 2008.

[26] H. Lu, W. Robert, G. Peirce, B. Ripa, and L. L. Baxter, “Comprehensive study of biomass particle combustion,” *Energy and Fuels*, vol. 22, no. 4, pp. 2826–2839, 2008.

[27] H. Thunman, F. Niklasson, F. Johnsson, and B. Leckner, “Composition of volatile gases and thermochemical properties of wood for modeling of fixed or fluidized beds,” *Energy and Fuels*, vol. 15, no. 6, pp. 1488–1497, 2001.

[28] S. Sinha, A. Jhalani, M. R. Ravi, and A. Ray, “Modeling of pyrolysis in wood: a review,” *SESI*, vol. 1, no. 10, pp. 41–62, 2000.

[29] N. Abani and A. F. Ghoniem, “Large eddy simulation of coal gasification in an entrained flow gasifier,” *Fuel*, no. 104, pp. 664–680, 2013.

[30] D. M. Christ, “The Effect of Char Kinetics on the Combustion of Pulverized Coal under Oxyfuel Conditions,” Dissertation, RWTH Aachen University, 2013.

[31] K. L. Smith, L. D. Smoot, T. H. Fletcher, and R. J. Pugmire, *The structure and reaction processes of coal*. N: Springer International Publishing, 1994.

[32] T. Jurena, “Numerical Modelling of Grate Combustion,” Dissertation, Brno University of Technology, 2012.

[33] Y. Haseli, J. A. van Oijen, and L. P. H. de Goey, “A detailed one-dimensional model of combustion of a woody biomass particle,” *Bioresour. Technol.*, vol.

102, no. 20, pp. 9772–9782, 2011.

[34] N. Fernando and M. Narayana, “A comprehensive two dimensional Computational Fluid Dynamics model for an updraft biomass gasifier,” *Renew. Energy*, vol. 99, pp. 698–710, 2016.

[35] R. E. Treybal, *Mass-Transfer operations*, 3rd ed. Singapore: McGraw-Hill, 1981.

[36] M. Frenklach, T. Bowman, and G. Smith, “GRI-Mech,” *Gas Technology Institute (GTI)*. [Online]. Available: http://combustion.berkeley.edu/gri-mech/index.htm.

[37] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. USA: Taylor & Francis, 1980.

[38] H. Liu, “CFD Modeling of Biomass Gasification Using a Circulating Fluidized Bed Reactor,” Dissertation, University of Waterloo, 2014.

[39] J. Kramb, J. Konttinen, A. Gómez-Barea, A. Moilanen, and K. Umeki, “Modeling biomass char gasification kinetics for improving prediction of carbon conversion in a fluidized bed gasifier,” *Fuel*, vol. 132, pp. 107–115, 2014.

[40] M. Kumar and A. F. Ghoniem, “Multiphysics simulations of entrained flow gasification. Part I: Validating the Nonreacting Flow Solver and the Particle Turbulent Dispersion Model,” *Energy Fuels*, vol. 26, no. 1, pp. 464–479, 2012.

[41] J. Sodja, “Turbulence models in CFD,” University of Ljubljana, 2007.

[42] H. E. Tahry, “k-ε Equation for Compressible Reciprocating Engine Flows.pdf,” *Energy*, vol. 7, pp. 345–353, 1983.

[43] P. Spalart, “The uses of DES: natural, extended and improper,” 2005.

[44] M. L. Shur, P. R. Spalart, M. K. Strelets, and A. K. Travin, “An Enhanced Version of DES with Rapid Transition from RANS to LES in Separated

Flows," *Flow, Turbul. Combust.*, vol. 95, no. 4, pp. 709–737, Dec. 2015.

[45] S. Gomez, "Changes and Settings for Standard Turbulence Model Implementation in OpenFOAM," New Mexico, 2006.

[46] F. R. Menter, M. Kuntz, and R. Langtry, "Ten Years of Industrial Experience with the SST Turbulence Model," *Turbul. Heat Mass Transf. 4*, vol. 4, pp. 625–632, 2003.

[47] A. P. Robinson, H. C. Coote, and P. Reupke, "Report on a Visit to Sri Lanka to carry-out a techno-economic appraisal of the NRI rice husk burner in collaboration with the Rice Processing Research and Development Centre," UK, 1991.

[48] X. Ku, T. Li, and T. Løvås, "CFD-DEM simulation of biomass gasification with steam in a fluidized bed reactor," *Chem. Eng. Sci.*, vol. 122, pp. 270–283, 2015.

[49] Y. Tsuji, T. Kawaguchi, and T. Tanaka, "Discrete particle simulation of two-dimensional fluidized bed," *Powder Technol.*, vol. 77, no. 1, pp. 79–87, 1993.

[50] B. H. Xu and A. B. Yu, "Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics," *Chem. Eng. Sci.*, vol. 52, no. 16, pp. 2785–2809, 1997.

[51] T. Song, J. Wu, L. Shen, and J. Xiao, "Experimental investigation on hydrogen production from biomass gasification in interconnected fluidized beds," *Biomass and Bioenergy*, vol. 36, pp. 258–267, 2012.

[52] D. G. C. Wickramasinghe, M. Narayana, and A. D. U. S. Amarasinghe, "Eulerian-Lagrangian Approach for Modeling of Biomass Fluidized Bed Combustion," in *Vidulka: National Energy Symposium*, 2017, pp. 209–213.

Appendix A: Publications

1. D. G. C Wickramasinghe, M. Narayana, and A. D. U. S. Amarasinghe, "Numerical Simulation of Suspension Biomass Combustor with Two Chambers," in *2018 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, 2018, pp. 226–230. doi: 10.1109/MERCon.2018.8421947, https://ieeexplore.ieee.org/abstract/document/8421947/
2. D. G. C Wickramasinghe, M. Narayana, and A. D. U. S. Amarasinghe, "Eulerian-Lagrangian Approach for Modeling of Biomass Fluidized Bed Combustion," in *Vidulka: National Energy Symposium*, SL, 2017, pp. 209–213.

Appendix B: CFD Code

1. Main loop

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2018 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Application
    MPPICFoam + coalChemistryFoam
    Diffusion corrective velocity

Description
    Transient solver for compressible, turbulent flow (LES) with a reacting,
    multiphase particle cloud and an inert particle cloud.

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "collidingRTurbulenceModel.H"
#include "basicReactingMultiphaseCloud.H"
#include "basicThermoCloud.H"
#include "psiReactionThermo.H"
#include "CombustionModel.H"
#include "radiationModel.H"
#include "SLGThermo.H"
#include "fvOptions.H"
#include "pimpleControl.H"
#include "pressureControl.H"
#include "localEulerDdtScheme.H"
#include "fvcSmooth.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "postProcess.H"

    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H"
    #include "createTimeControls.H"
    #include "createFields.H"
    #include "createFieldRefs.H"
    #include "initContinuityErrs.H"

    turbulence->validate();

    #include "compressibleCourantNo.H"
```

```
#include "setInitialDeltaT.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
    #include "readTimeControls.H"
    #include "compressibleCourantNo.H"
    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    lime.evolve();
    parcelsi.evolve();

    if (pimple.nCorrPimple() <= 1)
    {
        #include "rhoEqn.H"
    }

    // Update continuous phase volume fraction field
    alphac = max(1.0 - parcelsi.theta() - lime.theta(), alphacMin);
    alphac.correctBoundaryConditions();

    alphacf = fvc::interpolate(alphac);
    alphacPhi = alphacf*phi;
    alphacRho = alphac*rho;

    fvVectorMatrix cloudSU(lime.SU(U) + parcelsi.SU(U));    // N/m^3
    volVectorField cloudVolSUSu
    (
        IOobject
        (
            "cloudVolSUSu",
            runTime.timeName(),
            mesh
        ),
        mesh,
        dimensionedVector
        (
            "0",
            cloudSU.dimensions()/dimVolume,
            Zero
        ),
        zeroGradientFvPatchVectorField::typeName
    );

    cloudVolSUSu.primitiveFieldRef() = -cloudSU.source()/mesh.V();
    cloudVolSUSu.correctBoundaryConditions();
    cloudSU.source() = Zero;

    // --- PIMPLE loop
    while (pimple.loop())
    {
        #include "UEqn.H"
        #include "YEqn.H"
        #include "EEqn.H"

        // --- Pressure corrector loop
        while (pimple.correct())
        {
            #include "pEqn.H"
        }

        if (pimple.turbCorr())
        {
            turbulence->correct();
        }
```

```
        }

        rho = thermo.rho();

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End" << endl;

    return 0;
}


// ************************************************************************* //
```

2. createFields.H

```
#include "createRDeltaT.H"
#include "readGravitationalAcceleration.H"

IOdictionary physicalProperties
(
    IOobject
    (
        "physicalProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ_IF_MODIFIED,
        IOobject::NO_WRITE
    )
);
const dictionary& MWDict = physicalProperties.subDict("MW");
const dictionary& LeDict = physicalProperties.subDict("Le");
const dictionary& SigmaDict = physicalProperties.subDict("Sigma");
const dictionary& OmegaCoeffDict = physicalProperties.subDict("OmegaCoeff");

Info<< "Reading thermophysical properties\n" << endl;
autoPtr<psiReactionThermo> pThermo(psiReactionThermo::New(mesh));
psiReactionThermo& thermo = pThermo();
thermo.validate(args.executable(), "h", "e");

SLGThermo slgThermo(mesh, thermo);

basicSpecieMixture& composition = thermo.composition();
PtrList<volScalarField>& Y = composition.Y();

const word inertSpecie(thermo.lookup("inertSpecie"));
if (!composition.species().found(inertSpecie))
{
    FatalIOErrorIn(args.executable().c_str(), thermo)
        << "Inert specie " << inertSpecie << " not found in available species "
        << composition.species()
        << exit(FatalIOError);
}

PtrList<volScalarField> Diffs;
PtrList<volScalarField> Omega;
PtrList<dimensionedScalar> MW(Y.size());
PtrList<dimensionedScalar> Sigma(Y.size());
PtrList<dimensionedScalar> Om(Y.size());
PtrList<dimensionedScalar> Le(Y.size());
PtrList<label> speciesNames(Y.size());
```

```
volScalarField& p = thermo.p();

multivariateSurfaceInterpolationScheme<scalar>::fieldTable fields;

forAll(Y, i)
{
    fields.add(Y[i]);

    word nameDiffs("Diffs" + Y[i].name());  // Same as Y[i] index
    Diffs.append
    (
    new volScalarField
    (
        IOobject
        (
            nameDiffs,
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        mesh,
        dimensionedScalar(nameDiffs, dimensionSet(1,-1,-1,0,0,0,0), 0)  // D*rho
    )
    );

    word nameOmega("Omega" + Y[i].name());  // Same as Y[i] index
    Omega.append
    (
    new volScalarField
    (
        IOobject
        (
            nameOmega,
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        mesh,
        dimensionedScalar(nameOmega, dimless, 0)
    )
    );

    MW.set  // Same as Y[i] index
    (
        i,
        new dimensionedScalar(MWDict.lookup(Y[i].name()))
    );

    Le.set  // Same as Y[i] index
    (
        i,
        new dimensionedScalar(LeDict.lookup(Y[i].name()))
    );

    Sigma.set
    (
        i,
        new dimensionedScalar(SigmaDict.lookup(Y[i].name()))
    );

    Om.set
    (
        i,
        new dimensionedScalar(OmegaCoeffDict.lookup(Y[i].name()))
    );
}
fields.add(thermo.he());

dimensionedScalar DiffsD ("DiffsD", dimensionSet(1,1,-3,-1.5,0,0,0), 1);
dimensionedScalar cellVolume ("cellVolume", dimVolume, 3e-7);
```

```
const scalar AA = 1.06036;
const scalar BB = 0.15610;
const scalar CC = 0.19300;
const scalar DD = 0.47635;
const scalar EE = 1.03587;
const scalar FF = 1.52996;
const scalar GG = 1.76474;
const scalar HH = 3.89411;

volScalarField rho
(
    IOobject
    (
        "rho",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    thermo.rho()
);

volScalarField mu
(
    IOobject
    (
        "mu",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    thermo.mu()
);

Info<< "\nReading field U\n" << endl;
volVectorField U
(
    IOobject
    (
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

#include "compressibleCreatePhi.H"

mesh.setFluxRequired(p.name());

Info << "Creating field alphac\n" << endl;
volScalarField alphac
(
    IOobject
    (
        "alphac",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh,
    dimensionedScalar("alphac", dimless, 0)
);

volVectorField Vc
(
    IOobject
```

```
    (
        "Vc",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh,
    dimensionedVector("Vc", dimensionSet(1,-2,-1,0,0,0,0), vector::zero)
);

#include "createClouds.H"

// Dimension correction
scalar Diffcon(readScalar(physicalProperties.lookup("Diffcon")));

// Particle fraction upper limit
scalar alphacMin
(
    1.0 - readScalar(physicalProperties.lookup("alphaMax"))
);

// Update alphac from the particle locations
alphac = max(1.0 -lime.theta() - parcelsi.theta(), alphacMin);
alphac.correctBoundaryConditions();

surfaceScalarField alphacf("alphacf", fvc::interpolate(alphac));

// Volumetric continuous phase flux
surfaceScalarField alphacPhi("alphacPhi", alphacf*phi);

volScalarField alphacRho("alphacRho", alphac*rho);

Info<< "Creating turbulence model\n" << endl;
autoPtr<compressible::myturbulenceModel> turbulence
(
    compressible::myturbulenceModel::New
    (
        alphac,
        rho,
        U,
        alphacPhi,
        phi,
        thermo
    )
);

Info<< "Creating combustion model\n" << endl;
autoPtr<CombustionModel<psiReactionThermo>> combustion
(
    CombustionModel<psiReactionThermo>::New(thermo, turbulence())
);

Info<< "Creating field dpdt\n" << endl;
volScalarField dpdt
(
    IOobject
    (
        "dpdt",
        runTime.timeName(),
        mesh
    ),
    mesh,
    dimensionedScalar("dpdt", p.dimensions()/dimTime, 0)
);

Info<< "Creating field kinetic energy K\n" << endl;
volScalarField K("K", 0.5*magSqr(U));

volScalarField Qdot
(
    IOobject
```

```
    (
        "Qdot",
        runTime.timeName(),
        mesh,
        IOobject::READ_IF_PRESENT,
        IOobject::AUTO_WRITE
    ),
    mesh,
    dimensionedScalar("Qdot", dimEnergy/dimVolume/dimTime, 0.0)
);

#include "createMRF.H"
#include "createRadiationModel.H"
#include "createFvOptions.H"
```

3. createClouds.H

```
Info<< "\nConstructing reacting cloud" << endl;
basicReactingMultiphaseCloud parcelsi
(
    "reactingCloud1",
    rho,
    U,
    g,
    slgThermo
);

Info<< "\nConstructing limestone cloud" << endl;
basicThermoCloud lime
(
    "inertCloud1",
    rho,
    U,
    g,
    slgThermo
);
```

4. rhoEqn.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2015 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.
```

```
    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Global
    rhoEqn

Description
    Solve the continuity for density.

\*---------------------------------------------------------------------------*/

{
    fvScalarMatrix rhoEqn
    (
        fvm::ddt(alphac, rho)
      + fvc::div(alphacPhi)
      ==
        alphac*parcelsi.Srho(rho)
      + fvOptions(rho)
    );

    rhoEqn.solve();

    fvOptions.correct(rho);
}

// ************************************************************************* //
```

5. UEqu.H

```
    fvVectorMatrix UEqn
    (
        fvm::ddt(alphac, rho, U) + fvm::div(alphacPhi, U)
      + turbulence->divDevRhoReff(U)
     ==
        cloudSU
    );

    UEqn.relax();

    fvOptions.constrain(UEqn);

    volScalarField rAU(1.0/UEqn.A());
    surfaceScalarField rhorAUf("rhorAUf", fvc::interpolate(rho*rAU));
    surfaceScalarField alphacrhorAUf("alphacrhorAUf", alphacf*rhorAUf);

    surfaceScalarField phicForces   // kg/s
    (
       fvc::flux(rAU*cloudVolSUSu*rho)
        +        alphacf*rhorAUf*fvc::interpolate(rho)*(g & mesh.Sf())
    );

    if (pimple.momentumPredictor())
    {
        solve
        (
            UEqn
         ==
            fvc::reconstruct
            (
                phicForces/rhorAUf
              - fvc::snGrad(p)*mesh.magSf()
            )
        );
```

```
        fvOptions.correct(U);
        K = 0.5*magSqr(U);
    }
```

6. YEqn.H

```
tmp<fv::convectionScheme<scalar>> mvConvection
(
    fv::convectionScheme<scalar>::New
    (
        mesh,
        fields,
        phi,
        mesh.divScheme("div(phi,Yi_h)")
    )
);

volScalarField contErr("contErr", fvc::ddt(alphac, rho) + fvc::div(alphacPhi));

Vc = dimensionedVector("Vc", dimensionSet(1,-2,-1,0,0,0,0), vector::zero);

forAll(Y, i)
{
    Omega[i] =
        (AA/pow(T/Om[i],BB)) + (CC/exp(DD*T/Om[i])) + (EE/exp(FF*T/Om[i])) +
GG/exp(HH*T/Om[i]);
    Diffs[i] = DiffsD*rho*188.29225*Diffcon*pow(T, 1.5)*MW[i]/(Sigma[i]*p*Omega[i]);
    Vc += Diffs[i]*fvc::grad(Y[i], "grad(YiVc)");
}
Vc.correctBoundaryConditions();
// Volumetric correction velocity flux
surfaceScalarField Vcf("Vcf", fvc::interpolate(alphac*Vc, "interpolate_Vcf") &
mesh.Sf());

{
    combustion->correct();
    Qdot = combustion->Qdot();
    volScalarField Yt(0.0*Y[0]);

    forAll(Y, i)
    {
        if (i != inertIndex && composition.active(i))
        {
            volScalarField& Yi = Y[i];

            fvScalarMatrix YiEqn
            (
                fvm::ddt(alphac, rho, Yi)
              + mvConvection->fvmDiv((alphacPhi + Vcf), Yi)
              - fvm::laplacian(alphacf*(fvc::interpolate(Diffs[i])), Yi)
              ==
               alphac*combustion->R(Yi)
             + alphac*parcelsi.SYi(i, Yi)
            );

            YiEqn.relax();

            fvOptions.constrain(YiEqn);

            YiEqn.solve(mesh.solver("Yi"));

            fvOptions.correct(Yi);

            Yi.max(0.0);
```

```
            Yt += Yi;
        }
    }
    Y[inertIndex] = scalar(1) - Yt;
    Y[inertIndex].max(0.0);
}
```

7. EEqn.H

```
{
    volScalarField& he = thermo.he();

    fvScalarMatrix EEqn
    (
        fvm::ddt(alphac, rho, he) + mvConvection->fvmDiv(alphacPhi, he)
      + fvc::ddt(alphac, rho, K) + fvc::div(alphacPhi, K)
      - alphac*dpdt
      - fvm::laplacian(alphac*turbulence->alphaEff(), he)
     ==
        alphac*Qdot
      + alphacRho*(U&g)
      + alphac*lime.Sh(he)
      + alphac*parcelsi.Sh(he)
      + alphac*radiation->Sh(thermo, he)
      + fvOptions(rho, he)
    );

    EEqn.relax();

    fvOptions.constrain(EEqn);

    EEqn.solve();

    fvOptions.correct(he);

    thermo.correct();
    radiation->correct();

    Info<< "T gas min/max   = " << min(T).value() << ", "
        << max(T).value() << endl;
}
```

8. pEqn.H

```
    rho = thermo.rho();

    volVectorField HbyA(constrainHbyA(rAU*UEqn.H(), U, p));

    surfaceScalarField phiHbyA
    (
        "phiHbyA",
        (
            fvc::flux(rho*HbyA)
          + alphacrhorAUf*fvc::ddtCorr(rho, U, phi)
        )
    );

    phiHbyA += phicForces;
```

```
    // Update the pressure BCs to ensure flux consistency
    constrainPressure(p, rho, U, phiHbyA, rhorAUf, MRF);

    while (pimple.correctNonOrthogonal())
    {
        fvScalarMatrix pEqn
        (
            fvm::laplacian(alphacf*rhorAUf, p)
          + alphac*parcelsi.Srho()
         ==
            alphac*psi*correction(fvm::ddt(p))
          + fvc::div(alphacf*phiHbyA)
          + fvOptions(psi, p, rho.name())
        );

        pEqn.solve(mesh.solver(p.select(pimple.finalInnerIter())));

        if (pimple.finalNonOrthogonalIter())
        {
            phi = (phiHbyA - pEqn.flux()/alphacf);
            U = HbyA + rAU*fvc::reconstruct((phicForces -
pEqn.flux()/alphacf)/rhorAUf);
            U.correctBoundaryConditions();

        }
    }

    #include "rhoEqn.H"
    #include "compressibleContinuityErrs.H"

    fvOptions.correct(U);
    K = 0.5*magSqr(U);

    if (thermo.dpdt())
    {
        dpdt = fvc::ddt(p);
    }
```