

Modelling Turbulent Non-Premixed Combustion in Industrial Furnaces

Using the Open Source Toolbox OpenFOAM

by

Ali Hussain Kadar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 25, 2015 at 10:00 AM.



Student number: 4411056
Project duration: September 15, 2014 – August 25, 2015
Thesis committee: Dr. D. J. P. Lahaye, TU Delft, supervisor
Prof. dr. ir. C. Vuik, TU Delft
Dr. ir. M. Keijzer TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This work is the final part of obtaining a Master's degree in the Applied Mathematics department at the faculty of Electrical Engineering, Mathematics and Computer Science at the Delft University of Technology. This report presents and demonstrates the work of my thesis study performed under the supervision of Dr. D. J. P. Lahaye.

Combustion is a complex subject, requiring knowledge of many sub-disciplines such as physics, chemistry, thermodynamics and fluid mechanics. Although theories and empirical models for turbulent combustion have existed in literature since many decades, their practical use was restricted by the limited computational resources available. With the state of the art computational resources and advanced CFD techniques now at our disposal it is possible to optimize the design of combustion equipment's for improved efficiency and low pollutant emissions. With this work I hope to contribute towards a better understanding of turbulent combustion by making use of numerical simulation tools and techniques.

I wish to thank my supervisor Dr. D. J. P. Lahaye for the many interesting ideas, discussions and the wonderful excursions to the industrial plants of Almatris B.V and Tollenaar Machinebouw & Engineering B.V. during the course of the thesis. I wish to thank the thesis committee members Prof. dr. ir. C. Vuik and Dr. ir. M. Keijzer for the important feedback and evaluation of the masters thesis. I wish to thank Dr.ir. D.R. van der Heul for the help with OpenFOAM and many open source tools that I have used in my thesis. I wish to thank Prof. dr. D.J.E.M. Roekaerts and his team for bringing all the combustion enthusiasts together during the TU Delft combustion meetings and providing us opportunity to present our ideas. I am thankful to the M.Sc students in our masters program COSSE for the good time that we spent together working on our projects. Last but not least, I am grateful to my mother and fiancée for making sure every day that I have all my meals.

*Ali Hussain Kadar
Delft, August 2015*

Abstract

Measuring the temperature distribution inside the rotary kilns, using thermocouples for instance has proven to be difficult due to the harsh operating conditions of the kiln. Numerical modelling of turbulent combustion and the associated physical phenomenon thus proves to be an indispensable tool towards predicting the kiln operating conditions.

The purpose of the present work is to make a step towards modelling the cement rotary kiln used by Almatix B.V. in Rotterdam for the production of calcium-aluminate cement. The detailed mathematical model of the rotary kiln would be developed using OpenFOAM. OpenFOAM is a free, open source CFD software package capable of simulating complex fluid flows involving turbulence, chemical reactions and heat transfer. The OpenFOAM toolbox has detailed validation and verification studies for several well-known in the literature combustion test cases [1, 2, 3]. The main advantage of OpenFOAM is that, contrary to most of the commercial CFD software, it is license fee free and allows access to the source code, which is also the motivation behind this work.

To accurately model the Almatix kiln the following important phenomenon have to be taken into account: turbulent non-premixed combustion of hydrocarbon gases in the burner, radiative heat transfer distribution in the kiln and, the conjugate heat transfer through the furnace walls.

In the present work the new solver implemented in OpenFOAM for turbulent combustion and radiation modelling was validated using the benchmark Sandia Flame D test case [4]. There is good agreement seen between the results from simulations and experimental data for the Sandia Flame D test case indicating the adequacy and accuracy of the implemented transient solver and its readiness for further combustion application development. Due to the very complex geometry of the Almatix Kiln the relatively simple geometry of the Burner Flow Reactor (BFR) [5] is considered for further simulations. The simulation results obtained for the Burner Flow Reactor (BFR) are compared with the commercial package ANSYS Fluent for consistency. The OpenFOAM toolbox is evaluated in two stages of increasing complexity: isothermal(cold) flow simulation and non-premixed gas combustion simulation using a turbulent incompressible flow solver. The cold flow comparison gave almost identical results for both OpenFOAM and ANSYS Fluent. However the reacting flow results showed varying agreement with ANSYS Fluent. The mass frac-

tion of species showed good agreement but the temperature profile showed some deviations.

With more stringent global NO_x emission standards, predicting NO_x formation in industrial furnaces is now a priority. The CFD modelling of pollutant NO_x formation is considered in the present work. A new solver in OpenFOAM is developed for thermal NO prediction. The solver is validated with the ANSYS Fluent NO_x post-processing utility using the Burner Flow Reactor geometry. The effectiveness of NO_x reduction mechanisms including the variation of air to fuel equivalence ratio and flue gas re-circulation (FGR) is demonstrated using the Burner Flow Reactor test case.

From our study we conclude that OpenFOAM is a promising toolbox for modelling turbulent combustion and can be used for predicting the operating conditions of complex industrial furnaces. The current bottleneck identified with OpenFOAM is the very high computational cost of the implemented transient solver for turbulent combustion and radiation modelling. The computational cost of the transient solver far exceeds that of the steady state solvers available in commercial packages for example ANSYS Fluent. Therefore, to simulate very large scale industrial furnaces such as the Almatris Kiln in realizable time the implementation of a steady state solver for turbulent combustion applications in OpenFOAM is indispensable. It would also be essential to include the accompanying phenomenon of conjugate heat transfer into the solver. These can be accomplished as a part of the future work.

Contents

1	Introduction	10
1.1	Rotary Kilns	10
1.2	Computational Fluid Dynamics Modelling of Rotary Kilns	11
1.3	Objectives	12
1.4	Structure of this Report	13
2	Modelling Turbulent Non-Premixed Combustion	16
2.1	Introduction	16
2.2	Governing Equations for Turbulent Flows of Incompressible Fluids	17
2.2.1	Modelling the Reynolds Stresses	19
2.3	Governing Equations for Laminar Reacting Flows	21
2.3.1	Continuity and Momentum Equations	22
2.3.2	Species Transport Equations	22
2.3.3	Source Term for the Production/Consumption of Chemical Species	23
2.3.4	Energy Transport Equation	25
2.4	Favre-averaged Equations for Turbulent Reacting Flows	26
2.5	Modelling the Mean Reaction Rate	30
2.5.1	Eddy Break-Up Model	30
2.5.2	Eddy Dissipation Concept	32
3	Radiative Heat Transfer	34
3.1	Introduction	34
3.2	Radiative Transfer Equation	35
3.3	P1 Approximation for Radiative Heat Transfer Problems	37
3.3.1	Limitations of P1 Approximation	38
4	External Wall Heat Transfer	40
4.1	Introduction	40
4.2	1D Approximation for External Wall Heat Transfer	42
5	NO_x Formation	44
5.1	Introduction	44
5.2	Sources of NO Formation	44

5.2.1	Thermal NO Formation	45
5.3	Calculating NO Mass Fraction	48
6	Implementation in OpenFOAM	50
6.1	Introduction	50
6.2	OpenFOAM	50
6.2.1	OpenFOAM Structure	52
6.3	Solver for Thermal NO Prediction - NOxFoam	55
6.4	Solver for Turbulent Isothermal Flows - simpleFoam	61
6.5	Solver for Turbulent, Reacting Flows + Radiation - furnaceFoam	64
7	Sandia Flame D Validation	68
7.1	Introduction	68
7.2	Experimental Setup	69
7.3	Boundary and Initial Conditions	71
7.4	Computational Domain and Grid Independence Tests	72
7.5	Effect of Radiative Heat Transfer on the Temperature Distribution	74
7.6	Different Reaction Mechanisms for Methane Combustion	75
7.7	Comparison of Simulation Results with Experimental Data	79
7.8	Computational Time	80
7.9	Summary	82
8	Burner Flow Reactor	84
8.1	Introduction	84
8.2	Burner Flow Reactor Geometry	84
8.3	Boundary and Initial Conditions	86
8.4	Computational Domain and Mesh Non-orthogonality Tests	86
8.5	Conclusions	90
9	Cold Flow Computations	92
9.1	Introduction	92
9.2	Comparison of Results with ANSYS Fluent	92
9.3	Summary	93
10	Reacting Flow Computations	98
10.1	Introduction	98
10.2	Equivalence Ratio Calculations	98
10.3	Comparison of Results with ANSYS Fluent	99
10.4	Comparison of Infinitely fast chemistry model and PaSR model	103
10.4.1	2D Axi-symmetric Simulations	103
10.4.2	3D Simulations	104
10.5	Computational Time	105
10.6	Summary	108

11 NO_x Reduction Mechanisms	110
11.1 Validation of NO _x Foam using ANSYS Fluent	110
11.2 Thermal NO Reduction	114
11.2.1 Air to Fuel Equivalence Ratio Variation	115
11.2.2 Flue Gas Re-circulation (FGR)	119
11.3 Summary	120
12 Conclusions and Recommendations	122
12.1 Conclusions	122
12.2 Recommendations	123
Bibliography	126
A simpleFoam source code	132
B furnaceFoam source code	136
C System Properties	142

Chapter 1

Introduction

1.1 Rotary Kilns

Rotary kilns [6] are long cylindrical industrial furnaces inclined slightly to the horizontal as shown in fig. 1.1. They are used in a wide range of material processing industries of which cement is the most important. Over a billion tonnes of cement are produced every year, and cement kilns are the heart of this production process. Their capacity usually defines the capacity of the cement plant. As the main energy consuming, greenhouse gas and environmental pollutants emitting stage of cement manufacturing, improvement of kiln efficiency has been the central concern of cement manufacturing technology.

A typical cement rotary kiln consists of a long cylindrical tube made from steel, and lined with firebrick. The material to be processed is fed into the upper end of the cylinder, leaving a considerable amount of free-board or empty space. As the kiln rotates about its axis, the material bed gradually moves down towards the lower end, and undergoes a certain amount of stirring and mixing. At the lower end, fuel in the form of gas, oil, or pulverized solid fuel, is blown in through the burner pipe, producing a large concentric flame in the lower part of the kiln tube. As material moves under the flame, it reaches its peak temperature, before leaving the kiln at the lower end to be further processed. The heating of the kiln serves to drive the specific bed reactions, which, for either kinetic or thermodynamic reasons, require high temperatures [7].

The present work is a step towards modelling the rotary kiln used by Almatris B.V. a cement manufacturing company based in Rotterdam. The rotary kiln used by Almatris B.V. is a direct fired kiln in which the energy necessary to heat the material to the level required for the intended reactions is generated by the combustion of hydrocarbon fuels continuously fed to a burner placed in the free-board. In a direct fired rotary kiln, the heat is pushed through the interior of the kiln, meaning the heat transfer comes from the direct contact between the heat source and the material. Due to the high heat being pushed through the kiln, this kind of rotary kiln is refractory lined, usually with a brick

or castable lining, which protects the steel shell from the high internal temperatures. This kind of rotary kiln is a robust and efficient unit for heat transfer and pyroprocessing.

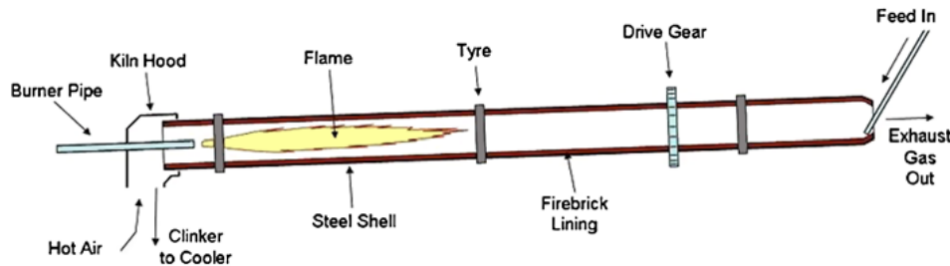


Figure 1.1: General layout of a cement rotary kiln.

1.2 Computational Fluid Dynamics Modelling of Rotary Kilns

The computational fluid dynamics modelling of a rotary kiln is a challenging task because of the many physical phenomenon simultaneously occurring in the kiln. The most important physical phenomenon that takes place in the kiln is the turbulent non-premixed combustion of the hydrocarbon gases injected from the burner. The heat released due to combustion results into very high temperatures (1500-3000K) in the flame zone close to the lower end of the kiln. The heat released is further convected upstream into the kiln due to turbulent flow. Due to very high temperatures in the kiln the radiative heat losses ($\propto T^4$) are significant and therefore needs to be accounted, for predicting accurate temperature distribution in the kiln. To protect the outer shell from the high internal temperatures, the lateral surface of the kiln is covered by an isolating lining of refractory material. The conjugate heat transfer through the furnace lining is another physical effect that needs to be included in the CFD model.

Various transport processes thus constitute the working principle of the kiln. To summarise, the physical effects that have to be taken into consideration for a sufficiently accurate mathematical model of the kiln are:

- Turbulent non-premixed combustion of hydrocarbon gases in the burner.
- Radiative thermal heat transfer distribution in the kiln.
- Conjugate heat transfer through the insulating furnace linings.

There are two important considerations towards modelling the Almatris kiln. First the burner injects fuel in axial and radial direction and is cooled by an amount of air forced through the secondary cooling slots. The second is that the material bed occupies no more

than five percent of the total volume of the kiln and therefore we may not take the material bed into account in our model and simulate an empty kiln.

The rotary kiln used by Almatris B.V. in Rotterdam has been analysed in a previous work [7] using the commercial package STAR-CCM+. The results obtained from these simulations were used to answer many important questions posed by the industry. Experimental validations at the plant later confirmed that the results from simulations were indeed very effective and resulted in hugely important cost savings to the industry. However there are many important questions which remain to be answered.

CFD modelling of the Almatris kiln is not a trivial task because of the very complex furnace geometry. The challenging aspect of this geometry is that the inlets of the burner are a factor thousand times smaller than the axial length of the kiln, imposing challenges in the mesh generation process. Therefore, to begin with, the relatively simple 2d axisymmetric geometry of the burner flow reactor is considered in the present work. Burner Flow Reactor geometry has been used in previous works for validating new CFD codes [8] [5].

1.3 Objectives

The objective of the present work is to develop a general multi-physics model for simulating complex industrial furnaces such as the Almatris kiln using the open source toolbox OpenFOAM. OpenFOAM offers a free advanced toolbox for solving complex physical problems and is equipped with the necessary libraries for modelling combustion, radiation and conjugate heat transfer. However, presently there exist no application in OpenFOAM that incorporates all the three physical phenomenon that are the basic ingredients of the general multi-physics model to be developed.

A challenging task would be to incorporate all the important physical effects into one application and to simulate Almatris kiln in realizable time making use of the parallel processing capabilities of OpenFOAM. However, due to time restrictions it was not possible to develop a CFD model incorporating all the physical effects and simulate Almatris kiln. The goal of the present work was therefore limited to developing a multi-physics model for turbulent combustion including radiative heat transfer. The inclusion of conjugate heat transfer effects into the model will be accomplished as part of the future work.

The objectives of the present work summarised are as follows

- To develop an application in OpenFOAM for turbulent combustion that includes radiative heat transfer.

- To develop an application in OpenFOAM for predicting pollutant Nitric Oxide (NO) formation in combustion devices.

1.4 Structure of this Report

In chapter 2 we begin by providing a brief introduction to incompressible turbulent flows in section 2.2. The need for turbulence modelling is illustrated and the governing equations for incompressible turbulent flows using the standard $k - \epsilon$ closure model for the Reynolds stresses are presented. To familiarize the reader with the subject of reacting flows, first the governing equations for laminar reacting flows including the chemical species and energy transport equations are discussed in section 2.3. The subsequent section 2.4 describes the set of partial differential equations governing turbulent reacting flows. The modelling of the kiln in realizable time requires resorting to a set of assumptions and simplified models for turbulence and chemistry. Therefore, the need of turbulence-chemistry interaction models is discussed in section 2.5 and the two most commonly used models for turbulence chemistry interaction namely the eddy break-up and eddy dissipation concept are illustrated.

In chapter 3 the radiative heat transfer equation is discussed with an emphasis on the P1-approximation model described in section 3.3. Chapter 4 illustrates the 1D external wall heat transfer boundary condition as a crude approximation to do away the need of solving a conjugate heat transfer problem. In chapter 5 the various sources of formation of air pollutant NO are discussed and the extended Zeldovich reaction mechanism governing thermal NO formation is described in detail.

Chapter 6 presents a brief overview of the OpenFOAM toolbox. The case setup in OpenFOAM is illustrated using the turbulent reacting flow case in fig. 6.2. The new solver **NOxFoam** implemented in OpenFOAM for thermal NO transport is described in detail in section 6.3. The objective is to familiarize the reader of the structure of OpenFOAM and the implementation of finite volume discretisation method using a basic solver. Due to time constraints and lack of documentation the more complex solver simpleFoam for turbulent isothermal flows and the new solver **furnaceFoam** derived from the existing solver reactingFoam for turbulent reacting flows are discussed in brief in section 6.4 and section 6.5 respectively.

The new solver furnaceFoam is validated using the benchmark Sandia Flame D [4] test case in chapter 7. Chapter 8 describes the geometry of the Burner Flow Reactor(BFR) and the computational mesh used for validating OpenFOAM results with the results obtained using the commercial software ANSYS Fluent.

Having defined the geometry of the test furnace and the computational mesh, we proceed further with cold flow and reacting flow simulations for the Burner Flow Reactor test case. The comparison of cold-flow and reacting-flow results with ANSYS Fluent are presented in chapter 9 and chapter 10 respectively. The validation of the new solver NOx-Foam for thermal NO prediction using ANSYS Fluent is presented in section 11.1. The effectiveness of NOx reduction mechanisms including the variation of air to fuel equivalence ratio and flue gas re-circulation (FGR) are demonstrated using the Burner Flow Reactor test case in section 11.2. Chapter 12 presents the conclusions of the thesis and future work that needs to be carried out to achieve the ambitious goal of simulating the Almatris Kiln using OpenFOAM technology.

Chapter 2

Modelling Turbulent Non-Premixed Combustion

2.1 Introduction

Turbulent combustion is one of the most important processes in engineering. It involves turbulent fluid flow, heat transfer, chemical reactions, radiative heat transfer and other complicated physical and chemical processes [9]. Combustion processes are governed by basic transport equations for fluid flow and heat transfer with additional models for turbulence, turbulence-chemistry interaction, radiative heat transfer and other important sub-processes.

Turbulence modelling is probably the most significant unresolved problem in classical physics [10]. Since the first preliminary turbulence model proposed by Ludwig Prandtl in the early 20th century called the mixing length model [11], the mechanism of turbulence generation has still not been fully understood. Since the flow is turbulent in nearly all engineering applications, the urgent need to resolve engineering problems has led to preliminary solutions called turbulence models. These models use systematic mathematical derivations based on the Navier–Stokes equations up to a certain point, but then they introduce closure hypotheses that rely on dimensional arguments and require empirical input [10].

Turbulent combustion is an intrinsically complex process because of the two-way coupling between flow and chemistry. Turbulence increases the mixing process and enhances combustion, and at the same time combustion releases heat. The heat released due to combustion generates flow instability through gas expansion and buoyancy, thus enhancing the transition to turbulence [10].

In this chapter we describe the governing equations for turbulent reacting flows. This chapter is divided into five sections including the introduction section itself. We begin by in-

roducing the various approaches DNS/LES/RANS for simulating turbulent incompressible fluids. We describe the ideas behind Reynolds-averaging and then state the governing equations for turbulent incompressible fluids obtained after applying the Reynolds-averaging to the laminar flow equations. In the third section we state the governing equations for laminar reacting flows including the chemical species transport and enthalpy equations along with the continuity and momentum equations. In the fourth section we introduce the concept of Favre-averaging and state the governing equations for turbulent reacting flows obtained after applying the Favre-averaging to the laminar reacting flow equations. In the last section we describe the two most commonly used models for closing the mean reaction rate term obtained after Favre-averaging of the species transport equation.

2.2 Governing Equations for Turbulent Flows of Incompressible Fluids

Turbulent flows involve various time and length scales ranging from the smallest dissipative scales (Kolmogorov microscales), up to the largest scale associated with the dimension of the problem. The various turbulence models available in literature vary in the extent to which they resolve these scales.

The simplest but computationally the most expensive approach is that of direct numerical simulation (DNS) where the Navier–Stokes equations are numerically solved without any turbulence model. This means that the whole range of spatial and temporal scales of the turbulence must be resolved using a very fine grid. Therefore, the computational cost of DNS is very high, even at low Reynolds numbers. For the Reynolds numbers encountered in most industrial applications, the computational resources required by a DNS far exceeds the capacity of the most powerful computers currently available thus making the DNS approach prohibitively expensive.

An alternative to the DNS approach is that of large eddy simulation (LES) where the small scales of the flow are not all resolved. This reduces the computational cost of the simulation. Large eddy simulation resolves large scales of the flow field solution allowing better fidelity than alternative approaches such as Reynolds-averaged Navier–Stokes (RANS) methods. It also models the small scales of the flow, rather than resolving them as direct numerical simulation (DNS) does. This makes the computational cost for practical engineering systems attainable using supercomputers. In contrast, direct numerical simulation, which resolves every scale of the solution, is prohibitively expensive for nearly all systems with complex geometry or flow configurations.

The most commonly used approach for turbulent industrial flows is the Reynolds-averaged Navier–Stokes (RANS) approach where an instantaneous quantity is decomposed into its

time-averaged and fluctuating quantities.

$$u(x, t) = \bar{u}(x) + u'(x, t), \quad \bar{u}(x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T u(x, t) dt \quad (2.1)$$

Here T is the averaging interval and must be large compared to the typical time scale of the fluctuations and u' is the fluctuation about the time averaged value. This idea of separating the flow variable (like velocity u) into the mean (time-averaged) component (\bar{u}) and the fluctuating component (u') was first proposed by Osborne Reynolds and is known as Reynolds averaging.

The instantaneous continuity and momentum equations governing the flow of an incompressible fluid are given by

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (2.2)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right], \quad i = 1, 2, 3 \quad (2.3)$$

where ρ is the density, p is the pressure, ν is the kinematic viscosity of the fluid and u_j the Cartesian velocity component in the x_j direction. For obtaining solutions to laminar flows the above equations can be solved by numerical discretization on not very fine grids, but for turbulent flows at high Reynolds number the grid has to be extremely fine to resolve all the eddies ranging from the smallest scales (Kolmogorov micro scales), up to the largest scale associated with the dimension of the problem. However Reynolds averaging can be used if one is only interested in the mean flow profile which is the case for most industrial flow applications. Substituting the Reynolds averaged quantities in the above equations and taking the mean of the equations we obtain the following unsteady Reynolds-averaged Navier–Stokes equations (or URANS).

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (2.4)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \overline{u_i' u_j'} \right], \quad i = 1, 2, 3 \quad (2.5)$$

where the over bar denotes the ordinary Reynolds averaging.

The equations 2.4, 2.5 describe the mean flow of a turbulent incompressible fluid at constant temperature. It has to be noted that often there is not very clear distinction made between incompressible fluids ($\rho = \text{constant}$) and incompressible flows ($\frac{D\rho}{Dt} = 0 \Rightarrow \nabla \cdot \vec{U} = 0$) and the terms are often used interchangeable. Further it can be observed that the pressure term is not explicitly present in the continuity or momentum equations but only

its gradient is present in the momentum equation, therefore the absolute value of pressure is not important but only the changes of pressure in space are important for incompressible flows.

2.2.1 Modelling the Reynolds Stresses

The continuity 2.4 and momentum 2.5 equations can be solved for the variables u_1 , u_2 , u_3 and p . However the Reynolds stress terms $-\overline{u_i' u_j'}$ arising from the fluctuating velocity field in the Reynolds-averaged momentum equation 2.5 are unknown and therefore needs to be closed. The diagonal components of the stress tensor are normal stresses and the off-diagonal components are shear stresses. Because of the symmetry of the Reynolds stress tensor there are six independent elements of the tensor and therefore six more unknowns in a 3-dimensional flow. To close the system, i.e. to get equal number of unknowns and equations, we have to model the Reynolds stresses in some way.

There are several approaches available in literature [11] for closing the Reynolds stresses. These models primarily differ in there complexity and thus the computational cost. We have the Reynolds stress model(RAS), which is the most detailed model involving six partial differential equations for closing the six independent Reynolds stresses. We also have simplified two equation models e.x. $k - \epsilon$ involving two transport equations (PDEs) for the turbulent kinetic energy(k) and dissipation(ϵ). They are based upon the following turbulent viscosity hypothesis introduced by Boussinesq [12].

$$-\overline{\rho u_i' u_j'} = \mu_t \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (2.6)$$

where the turbulent kinetic energy(k) is defined as half the trace of the Reynolds stress tensor and μ_t is a scalar property called the turbulent eddy viscosity which is normally computed from the two transported variables.

The primary factors affecting the choice of turbulence model for industrial flows is the computational cost. The model should be reasonably accurate in modelling the physics encompassed in the flow and at the same time should be practically realizable. Since for most practical engineering applications the Boussinesq hypothesis perform very well [13], the additional computational expense of the Reynolds stress model is not justified. Therefore, for our application we consider the most commonly used model: standard $k - \epsilon$, a two equation model which gives a general description of turbulence by means of two transport equations (PDEs). Without giving detailed derivations we state the equations governing flow of an incompressible fluid using the Standard $k - \epsilon$ closure model for Reynolds stresses [14] . The following system of simultaneous equations are solved for the 6 unknowns p , $\overline{u_1}$, $\overline{u_2}$, $\overline{u_3}$, k and ϵ during the incompressible isothermal flow computation in chapter 9.

Continuity and Momentum Equations

The Reynolds-averaged continuity and momentum equations after closing the Reynolds stresses using the Boussinesq turbulent viscosity hypothesis are

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (2.7)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial}{\partial x_i} \left[\bar{p} + \frac{2}{3} \rho k \right] + \frac{\partial}{\partial x_j} \left[\nu_{eff} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right], \quad i = 1, 2, 3 \quad (2.8)$$

The effective kinematic viscosity ν_{eff} is computed as the sum of the laminar and the turbulent kinematic viscosity $\nu_{eff} = \nu + \nu_T$ where the turbulent kinematic viscosity ν_T is given by

$$\nu_T = C_\mu \frac{k^2}{\epsilon}$$

Transport equation for the turbulent kinetic energy(k)

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\nu_{eff}^k \frac{\partial k}{\partial x_j} \right] + \nu_T \frac{\partial \bar{u}_i}{\partial x_j} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \epsilon \quad (2.9)$$

The effective viscosity for turbulent kinetic energy ν_{eff}^k is given by

$$\nu_{eff}^k = \nu + \frac{\nu_T}{\sigma_k}$$

where σ_k is the turbulent Prandtl number for k

Transport equation for the dissipation(ϵ)

$$\frac{\partial \epsilon}{\partial t} + \bar{u}_j \frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\nu_{eff}^\epsilon \frac{\partial \epsilon}{\partial x_j} \right] + C_{1\epsilon} \frac{\epsilon}{k} \nu_T \frac{\partial \bar{u}_i}{\partial x_j} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - C_{2\epsilon} \frac{\epsilon^2}{k} \quad (2.10)$$

The effective viscosity for turbulent dissipation ν_{eff}^ϵ is given by

$$\nu_{eff}^\epsilon = \frac{\nu_T}{\sigma_\epsilon}$$

where σ_ϵ is the turbulent Prandtl number for ϵ

The modal constants are given by

$$\sigma_k = 1 \quad \sigma_\epsilon = 1.3 \quad C_\mu = 0.09 \quad C_{1\epsilon} = 1.44 \quad C_{2\epsilon} = 1.92$$

The default values of the standard $k - \epsilon$ model constants $C_{1\epsilon}$, $C_{2\epsilon}$, C_μ , σ_k , and σ_ϵ have been determined from experiments with air and water for fundamental turbulent shear flows including homogeneous shear flows and decaying isotropic grid turbulence. They have been found to work fairly well for a wide range of wall-bounded and free shear flows [13].

2.3 Governing Equations for Laminar Reacting Flows

There are many types of combustion processes example gaseous fuel combustion, liquid fuel combustion, spray combustion etc [9]. Gaseous fuel combustion involves chemical reactions between a fuel and an oxidant that are both in the gas phase. Further based upon the composition of the gaseous fuel under consideration there are two categories of gaseous combustion processes: premixed combustion and non-premixed combustion. In premixed combustion the fuel is mixed with air prior to combustion and the combustion occurs in the combustion chamber when the mixture is ignited. For example, combustion in a spark ignition internal combustion engine (petrol engine). However, in non-premixed combustion situations such as furnaces, the fuel and ambient air streams are introduced into the combustion chamber separately [9]. Non-premixed flames are also called diffusion flames because fuel and air are introduced to the combustion zone in two or more separate streams and are subsequently brought together due to diffusion and mixing prior to combustion. In this work we are primarily interested in non-premixed gaseous combustion and hereafter by combustion we would refer to non-premixed gaseous combustion.

Combustion is a complex subject requiring a considerable amount of knowledge and experience in the areas of thermodynamics, chemical equilibrium and chemical kinetics [15] besides the understanding of the physics of flow and turbulence. In the last section we stated the Reynolds-averaged continuity 2.2 and momentum 2.3 equations. These can be solved to obtain the unknown pressure and velocity fields for an incompressible fluid at constant temperature. However turbulent reacting flows are more complex and includes additional equations for the transport of energy and involved chemical species. The local density of the reacting mixture is not constant and is dependent on pressure, species concentrations and on the mixture temperature. Further since the properties of the fluid are a function of temperature, the continuity and momentum equations are coupled with the energy equation and therefore cannot be solved independently.

In this section we will describe the equations governing laminar reacting flows. These simultaneous non-linear partial differential equations can be solved numerically on a fine mesh for simulating laminar reacting flows. However for turbulent flows Favre-averaged equations describing the mean flow are solved for reasons similar to the need of Reynolds-averaging as illustrated in the previous section. The Favre-averaged equations for simulating turbulent reacting flows will be presented in the next section.

2.3.1 Continuity and Momentum Equations

The instantaneous continuity and momentum equations describing flow with varying density are given by:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \quad (2.11)$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \right], \quad i = 1, 2, 3 \quad (2.12)$$

where p is pressure, ρ is the average density of the mixture defined in the next section and u_j is the Cartesian velocity component in the x_j direction. The laminar viscosity μ is calculated using Sutherlands's law [16]

$$\mu = A_s \frac{T^{3/2}}{T + T_s}$$

where $A_s = 1.67212 \times 10^{-6}$, $T_s = 170.672$ are constants.

2.3.2 Species Transport Equations

During combustion chemical species react and form products and in the process heat is released due to the exothermic reactions. The heat released and the species formed are transported by the flow. Therefore, the basis for any combustion model is the continuum formulation of the balance equations for energy and the chemical species.

Let there are m chemical species $S_1, S_2 \dots S_m$ present in the mixture whose mass fractions are given by $Y_1, Y_2 \dots Y_m$. Applying the law of conservation of mass we obtain the following scalar transport equation for specie S_s .

$$\underbrace{\frac{\partial \rho Y_s}{\partial t}}_{\text{Transient term}} + \underbrace{\frac{\partial(\rho u_j Y_s)}{\partial x_j}}_{\text{Convection term}} = \underbrace{\frac{\partial}{\partial x_j} \left[\rho D_s \frac{\partial Y_s}{\partial x_j} \right]}_{\text{Diffusion term}} + \underbrace{\dot{\omega}_s}_{\text{Source term}} \quad s = 1, 2, \dots, m \quad (2.13)$$

where ω_s is the source term accounting for production/consumption of the specie s illustrated in the subsection 2.3.3, and D_s is the specie's diffusion coefficient with respect to an abundant specie, for instance N_2 . The molecular transport processes that cause diffusive fluxes are quite complicated and here we have considered the most elementary binary flux approximation in calculating the diffusion term. The binary flux approximation is a good approximation for turbulent diffusion flames where molecular transport processes are less important than turbulent transport processes [10].

Given the species mass fractions and pressure the local density of the mixture in the

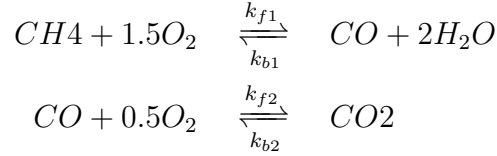
continuity 2.11 and momentum 2.12 equations can be calculated using the ideal gas law for the participating chemical species. The density of the mixture is then given by

$$\rho = \frac{pM_{avg}}{RT}$$

where $M_{avg} = \frac{1}{\sum_s \frac{Y_s}{M_s}}$ is the average molar mass of the mixture.

2.3.3 Source Term for the Production/Consumption of Chemical Species

To illustrate the computation of the source term for the production/consumption of a specie $\dot{\omega}_s$, let us consider the following two-step reaction mechanism of Westbrook and Dryer(1981) [17] for methane combustion.



where the forward rate constants k_{f1} , k_{f2} and the backward rate constants k_{b1} , k_{b2} are functions of temperature. Arrhenius equation gives the dependence of the rate constant k of a chemical reaction on the absolute temperature T (in kelvins), where A is the pre-exponential factor, E_a is the activation energy, and R is the universal gas constant.

$$k = A e^{-E_a/RT}$$

Let us consider the specie CO which is formed due to the forward reaction 1 and backward reaction 2 and consumed due to the forward reaction 2 and backward reaction 1. Therefore, the rate of change of CO due to reaction 1 and reaction2 is given by

$$\begin{aligned} \left. \frac{d[CO]}{dt} \right|_{r1} &= 1 \cdot \left(k_{f1}[CH_4]^1[O_2]^{1.5} - k_{b1}[CO]^1[H_2O]^2 \right) \\ \left. \frac{d[CO]}{dt} \right|_{r2} &= -1 \cdot \left(k_{f2}[CO]^1[O_2]^{0.5} - k_{b2}[CO_2]^1 \right) \end{aligned}$$

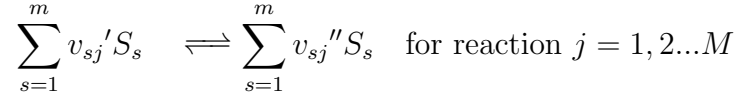
The net rate of change of CO is then given by

$$\begin{aligned} \left. \frac{d[CO]}{dt} \right|_{net} &= \left. \frac{d[CO]}{dt} \right|_{r1} + \left. \frac{d[CO]}{dt} \right|_{r2} \\ \left. \frac{d[CO]}{dt} \right|_{net} &= 1 \cdot \left(k_{f1}[CH_4]^1[O_2]^{1.5} - k_{b1}[CO]^1[H_2O]^2 \right) \\ &\quad - 1 \cdot \left(k_{f2}[CO]^1[O_2]^{0.5} - k_{b2}[CO_2]^1 \right) \end{aligned}$$

Similarly the rate of change of CH₄, O₂ and H₂O can be expressed as

$$\begin{aligned}\frac{d[CH_4]}{dt}\Big|_{net} &= -1 \cdot \left(k_{f1}[CH_4]^1[O_2]^{1.5} - k_{b1}[CO]^1[H_2O]^2 \right) \\ \frac{d[O_2]}{dt}\Big|_{net} &= -1.5 \cdot \left(k_{f1}[CH_4]^1[O_2]^{1.5} - k_{b1}[CO]^1[H_2O]^2 \right) \\ &\quad - 0.5 \cdot \left(k_{f2}[CO]^1[O_2]^{0.5} - k_{b2}[CO_2]^1 \right) \\ \frac{d[H_2O]}{dt}\Big|_{net} &= 2 \cdot \left(k_{f1}[CH_4]^1[O_2]^{1.5} - k_{b1}[CO]^1[H_2O]^2 \right)\end{aligned}$$

Therefore, we have a system of non-linear ODE's which must be integrated in time to obtain the concentration of the participating species CO, CH₄, O₂ and H₂O. For reacting systems in most industrial applications there are hundreds of species involved and can be even larger for more complex hydrocarbons. The simplest hydrocarbon widely used in industry is Methane. The detailed reaction mechanism of Methane involves 325 reactions and 53 species given by the GRI-3.0 mechanism [18]. To represent very large reaction mechanisms we use the following indexing notation for the reactions and the participating species.



and each reaction involving species $s = 1, 2, \dots, m$. The reaction rate for a specie s due to reaction j is given by

$$\dot{q}_{sj} = \frac{d[S_s]}{dt}\Big|_j = (v_{sj}'' - v_{sj}') \cdot \left[k_{fj} \prod_{s=1}^m [S_s]^{v_{sj}'} - k_{bj} \prod_{s=1}^m [S_s]^{v_{sj}''} \right]$$

The net reaction rate for specie s is therefore

$$\dot{q}_s = \sum_{j=1}^M \dot{q}_{sj}$$

Finally the source term in the species transport equation is given by

$$\dot{\omega}_s = M_s \dot{q}_s$$

It has to be noted that the source term in the species transport equation has to be evaluated during each solve iteration where the mass fractions of the species are computed from the new pressure, velocity and temperature distribution. Therefore, during each solve iteration the very large and stiff system of ODE's needs to be integrated which poses a computational challenge. To reduce the cost of computation generally a simple reaction mechanism describing the combustion of fuel is considered for most practical applications.

2.3.4 Energy Transport Equation

The heat released during combustion can be represented in different forms for obtaining the governing energy transport equation. The most common forms are internal energy(e) and enthalpy(h). The equations obtained using either of the two forms result into the same solution because they essentially represent the law of conservation of energy applied to the system. However, different fluids and classes of flow problems are solved optimally using a particular form of energy. For example, internal energy(e) is the preferred form of energy for liquids, and enthalpy(h) is preferable for problems involving gases, particularly combustion.

The enthalpy of the mixture can be expressed as the mass-weighted sum of the specific enthalpies h_s of species s

$$h = \sum_s Y_s h_s \quad h_s = h_s^0 + \int_{T_0}^T C_{p_s}(T) dT$$

where $C_{p_s}(T)$ is the specific heat capacity of specie s at constant pressure and temperature T . h_s^0 is the enthalpy of formation of specie s at standard temperature and pressure. The enthalpy of formation of various chemical species are available in the JANAF Thermochemical tables [19] and can be used in most combustion simulation codes in the form of standard libraries. The polynomial fits for the temperature dependence of the specific heat capacity of different species are also available in literature and can be imported for instance in CHEMKIN thermophysical data format [18].

Applying the principle of energy conservation the balance equation for the enthalpy of the system is given by.

$$\frac{\partial}{\partial t}(\rho h) + \frac{\partial}{\partial x_j}(\rho u_j h) = \frac{\partial}{\partial x_j} \left[\frac{\mu}{Pr_h} \frac{\partial h}{\partial x_j} + \mu \left(\frac{1}{Sc_s} - \frac{1}{Pr_h} \right) \sum_s h_s \frac{\partial Y_s}{\partial x_j} \right] + \frac{\partial p}{\partial t} + S_{rad} + S_h \quad (2.14)$$

Since we are primarily interested in turbulent flows in furnaces the terms describing viscous heating ($\nabla \cdot \tau \vec{u}$) and convective change of pressure ($\vec{u} \cdot \nabla p$) which are significant only at high speeds have not been considered because we are interested in the small Mach number limit. The transient pressure term $\frac{\partial p}{\partial t}$ can also be neglected in open flames where the pressure is approximately constant and equal to the static pressure [10]. S_{rad} in the enthalpy equation is the source term accounting for thermal radiation illustrated in chapter 3 and S_h is the source term due to combustion

In the enthalpy equation Sc_s is the species Schmidt number defined as

$$Sc_s = \frac{\mu}{\rho D_s}$$

for specie s and Pr_h is the mixture Prandtl number for enthalpy(h) defined as

$$Pr_h = \frac{\mu C_p}{k_c}$$

where k_c is the average thermal conductivity of the mixture and $Cp = \sum_s Y_s Cp_s$ is the specific heat capacity of the mixture at constant pressure. The Lewis number for a specie s is defined as

$$Le_s = \frac{k_c}{\rho D_s Cp} = \frac{Sc_s}{Pr_h}$$

Under the assumption of single diffusion coefficient i.e. $D_s = D$ and constant Lewis number of unity the enthalpy equation simplifies to

$$\frac{\partial}{\partial t}(\rho h) + \frac{\partial}{\partial x_j}(\rho u_j h) = \frac{\partial}{\partial x_j} \left[\frac{\mu}{Pr_h} \frac{\partial h}{\partial x_j} \right] + S_{rad} + S_h \quad (2.15)$$

which exactly resembles the scalar transport equation for species. However such oversimplifying assumptions may not always be accurate (and may in fact be quite inaccurate) but allows simplified simulations to predict the velocity, temperature and species distribution which can be used as first guess for more detailed simulations.

After solving the transport equation for enthalpy, the temperature in the domain can be updated as follows

$$T = \frac{h - Y_{fuel} h_{fuel}}{Cp} \quad \overline{Cp} = \frac{1}{T - T_{ref}} \int_{T_0}^T Cp dT \quad (2.16)$$

where h_{fuel} is the enthalpy of combustion of the fuel.

2.4 Favre-averaged Equations for Turbulent Reacting Flows

In the last section we described the instantaneous equations governing reacting flows. However, since the flow is turbulent in nearly all engineering applications and only the mean characteristics of the flow are of interest we average the instantaneous equations governing reacting flows to deduce the mean flow characteristics. In this section we state the Favre-averaged equations governing turbulent reacting flows which are solved during the turbulent reacting flow computations in chapter 10.

In section 2.2 we showed that Reynolds averaging 2.1 and modelling of the resulting averages of fluctuating product terms made it possible to predict the mean flow field governing incompressible turbulent flows. Similarly the mean flow field governing turbulent reacting flows can be obtained by suitable averaging of the governing equations for laminar flows 2.11-2.15 and modelling the resulting averages of fluctuating product terms obtained. However it is considerably more difficult than for incompressible flows because of the fact that strong and highly localised heat generation in reacting flows causes the density to vary as a function of position. There are also density fluctuations due to turbulence [9].

It is easy to demonstrate that the presence of density fluctuations gives rise to additional terms when Reynolds averaging is used [9]. The additional terms arise from correlations between the velocity and density fluctuations in a reacting flow and have to be modelled. Many terms of this type appear in the Reynolds-averaged momentum, scalar and species transport equations. To reduce the number of separate terms requiring modelling in reacting flows with variable density, we use a density-weighted averaging procedure known as Favre-averaging instead of the commonly used Reynolds averaging.

In Favre-averaging the density-weighted mean velocity is defined as $\tilde{u} = \frac{\overline{\rho u}}{\bar{\rho}}$. The instantaneous velocity u is written as $u = \tilde{u} + u''$. In contrast to the Reynolds decomposition, where u' represents a turbulent velocity fluctuation, the quantity u'' also includes effects of density fluctuations. We now present the Favre-averaged equations used to model turbulent reacting flows without giving detailed derivations.

The Favre-averaged continuity equation:

Introducing Favre-averages into the continuity equation 2.11 and taking density weighted mean we obtain the following Favre-averaged continuity equation.

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j) = 0 \quad (2.17)$$

where the overbar denotes the ordinary Reynolds averaging, while the tilde denotes mass weighted Favre-averages.

The Favre-averaged momentum equation:

Introducing Favre-averages into the momentum equation 2.12 and taking density weighted mean we obtain the following Favre-averaged momentum equation.

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_i \tilde{u}_j) = - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\bar{\tau}_{ij} - \overline{\rho u_i'' u_j''} \right) \quad (2.18)$$

where τ_{ij} is the viscous stress tensor.

The Favre-averaged species transport equations:

Introducing Favre-averages into the species transport equation 2.13 and taking density weighted mean we obtain the following Favre-averaged species transport equation.

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{Y}_s) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{Y}_s \tilde{u}_j) = \frac{\partial}{\partial x_j} \left(\bar{\rho} D_s \frac{\partial \tilde{Y}_s}{\partial x_j} - \overline{\rho Y_s'' u_j''} \right) + \tilde{\omega}_s, \quad s = 1, \dots, m$$

Applying gradient diffusion assumption

$$-\overline{\rho Y_s'' u_j''} = \frac{\mu_t}{Sc_s^t} \frac{\partial \tilde{Y}_s}{\partial x_j}$$

for modelling the averages of products of fluctuating quantities we obtain

$$\frac{\partial}{\partial t}(\overline{\rho \tilde{Y}_s}) + \frac{\partial}{\partial x_j}(\overline{\rho \tilde{Y}_s \tilde{u}_j}) = \frac{\partial}{\partial x_j} \left(\Gamma_s \frac{\partial \tilde{Y}_s}{\partial x_j} \right) + \tilde{\omega}_s, \quad s = 1, \dots, m \quad (2.19)$$

where $\tilde{\omega}_s$ is the Favre-averaged reaction rate discussed further in section 2.5. $\Gamma_s = \left(\frac{\mu}{Sc_s} + \frac{\mu_t}{Sc_s^t} \right)$ where μ_t and Sc_s^t are turbulent viscosity and turbulent Schmidt number for specie s respectively.

The Favre-averaged enthalpy equation:

Introducing Favre-averages into the enthalpy transport equation 2.15 and taking density weighted mean we obtain the following Favre-averaged enthalpy transport equation.

$$\frac{\partial}{\partial t}(\overline{\rho \tilde{h}}) + \frac{\partial}{\partial x_j}(\overline{\rho \tilde{h} \tilde{u}_j}) = \frac{\partial}{\partial x_j} \left(\overline{\rho \alpha} \frac{\partial \tilde{h}}{\partial x_j} - \overline{\rho h'' u_j''} \right) + \overline{S_{rad}} + \overline{S_h}$$

Applying gradient diffusion assumption

$$-\overline{\rho h'' u_j''} = \frac{\mu_t}{Pr_h^t} \frac{\partial \tilde{h}}{\partial x_j}$$

for modelling the averages of products of fluctuating quantities we obtain

$$\frac{\partial}{\partial t}(\overline{\rho \tilde{h}}) + \frac{\partial}{\partial x_j}(\overline{\rho \tilde{h} \tilde{u}_j}) = \frac{\partial}{\partial x_j} \left(\Gamma_h \frac{\partial \tilde{h}}{\partial x_j} \right) + \overline{S_{rad}} + \overline{S_h} \quad (2.20)$$

where $\Gamma_h = \left(\frac{\mu}{Pr_h} + \frac{\mu_t}{Pr_h^t} \right)$. Pr_h^t is the turbulent Prandtl number for enthalpy.

Closing the Reynolds stresses using the Standard $k - \epsilon$ turbulence model

In section 2.2.1 we stated the governing equations for incompressible turbulent flow where Standard $k - \epsilon$ model was used for closing the Reynolds stresses 2.6. Similarly the Standard $k - \epsilon$ model may be applied with Favre-averaged transport equations for k and ϵ to close the Reynolds stresses as follows

$$\overline{\rho u_i'' u_j''} = -\mu_t \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{u}_k}{\partial x_k} \right) + \frac{2}{3} \overline{\rho k} \delta_{ij}$$

where δ_{ij} is the Kronecker delta, $\mu_t = \frac{C_\mu \bar{\rho} \tilde{k}^2}{\tilde{\epsilon}}$ is the eddy viscosity, and C_μ is a model constant.

The equation for turbulence kinetic energy \tilde{k} :

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{k}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{k}) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{Pr_k^t} \right) \frac{\partial \tilde{k}}{\partial x_j} \right] + G - \bar{\rho}\tilde{\epsilon}$$

and the equation for dissipation rate of turbulence kinetic energy $\tilde{\epsilon}$:

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{\epsilon}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{\epsilon}) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{Pr_\epsilon^t} \right) \frac{\partial \tilde{\epsilon}}{\partial x_j} \right] + C_{\epsilon_1} \frac{\tilde{\epsilon}}{\tilde{k}} G - C_{\epsilon_2} \bar{\rho} \frac{\tilde{\epsilon}^2}{\tilde{k}}$$

where Pr_k^t and Pr_ϵ^t are the turbulent Prandtl numbers for k and ϵ and C_{ϵ_1} and C_{ϵ_2} are model constants, and the rate of turbulence kinetic energy production G is given as follows:

$$G = - \sum_{i=1}^3 \sum_{j=1}^3 \overline{\rho u_i'' u_j''} \frac{\partial \tilde{u}_i}{\partial x_j}$$

The default values for the model constants are given by

$$Pr_k^t = 1 \quad Pr_\epsilon^t = 1.3 \quad C_\mu = 0.09 \quad C_{\epsilon_1} = 1.44 \quad C_{\epsilon_2} = 1.92$$

2.5 Modelling the Mean Reaction Rate

In section 2.3.3 we illustrated how the source term $\dot{\omega}_s$ in the species transport equation eq. (2.13) governing laminar flows is computed using the chemical kinetics approach. However most industrial flows are turbulent and the Favre-averaged source term in the species transport equation eq. (2.19) therefore needs to be modeled. Favre-averaging of the species generation term $\dot{\omega}_s$ to obtain the Favre-averaged mean reaction rate $\tilde{\dot{\omega}}_s$ is the subject of many textbooks on Combustion Modelling [20] [9].

As the reaction rate is expressed by the Arrhenius law with Temperature in the exponent, it is a highly non-linear term. The mean reaction rate $\tilde{\dot{\omega}}_s$ therefore cannot be easily expressed as a function of the Favre-averaged mass fractions \tilde{Y}_s , the mean density $\bar{\rho}$ and mean temperature \bar{T} . To obtain the mean reaction rate $\tilde{\dot{\omega}}_s$ a Taylor series expansion for the exponential term may be used. However, it has been observed that the mean reaction rate $\tilde{\dot{\omega}}_s$ in this way involves correlations and many unknowns that have to be modelled [9]. When detailed reaction mechanisms involving many species and elementary reactions are involved, it is not possible to model these correlations. Therefore, a considerable amount of effort in turbulent combustion modelling has been directed towards the development of models which avoid the solution of the mean reaction rate term $\tilde{\dot{\omega}}_s$.

There are many models available in literature that have been implemented in practice [21] and are found to be accurate to different degrees based upon their complexity and the physics they represent. Turbulence chemistry interaction models can be broadly divided into two categories viz equilibrium chemistry models and detailed chemistry models [13]. Here we briefly describe the two most common turbulence chemistry interaction models used in industry, drawing heavily from the book by Versteeg and Malalasekera [9] and reviews by Magnussen [22].

2.5.1 Eddy Break-Up Model

One of the most widely used model in industry for modelling turbulent diffusion flames is the eddy break-up model by Spalding (1971). It is an equilibrium chemistry model based on the assumption that the reaction rate is controlled by turbulent mixing. The equilibrium chemistry assumption implies that the rate of chemical reaction is faster than the rate of mixing of fuel and oxygen, so that equilibrium is attained earlier than the residence time τ_{res} . Here residence time is defined as the average amount of time that the molecules will stay in a small control volume of the system before being transported away by convection/diffusion. Therefore, the equilibrium chemistry model is a good approximation when $\tau_{res} \gg \tau_{eq}$.

In combustion simulations, the Damkohler number, Da, is often used to assess the suitability of an equilibrium chemistry model as a replacement to the detailed chemistry

model. It is a dimensionless number used to compare the flow time scale and chemical time scale. It is defined as follows.

$$Da = \frac{\text{reaction rate}}{\text{convective mass transport rate}} \propto \frac{\tau_{res}}{\tau_{eq}}$$

If Da is large, it means that the chemistry has always the time to fully consume the fresh mixture and turn it into equilibrium. Real flames are usually close to this state. The characteristic reaction time, $(Ae^{-T_a/T_s})^{-1}$, is estimated of the order of the tenth of a ms [23]. When Da is high (≥ 10) the chemical species can be assumed to have attained equilibrium and therefore the computationally expensive Arrhenius chemical kinetics approach can be avoided. This makes the eddy Break-Up model significantly faster than the detailed chemical kinetics models and is therefore, the method of choice for industrial applications where the computational time is often the bottleneck.

To illustrate the concept we consider the following global reaction mechanism for combustion of a certain fuel.

- 1 Kg Fuel(fu) + w Kg Oxidant(ox) = (1+w) Kg Products(pr)

Then rate of consumption of fuel is specified as a function of local flow properties.

$$\widetilde{\dot{\omega}}_{fu} = -C_R \frac{\bar{\rho}}{\tau_{mix}} \widetilde{Y}_{fu}$$

where the mixing-controlled rate of reaction is expressed in terms of the turbulence mixing time $\tau_{mix} = \frac{\tilde{k}}{\epsilon}$. Similarly, the mixing-controlled rate of reaction for oxygen and products are expressed as follows.

$$\begin{aligned} \widetilde{\dot{\omega}}_{ox} &= -C_R \frac{\bar{\rho}}{\tau_{mix}} \frac{\widetilde{Y}_{ox}}{w} \\ \widetilde{\dot{\omega}}_{pr} &= -C_R' \frac{\bar{\rho}}{\tau_{mix}} \frac{\widetilde{Y}_{pr}}{1+w} \end{aligned}$$

Finally the rate of the reaction is taken as the minimum of all the reaction rates.

$$\widetilde{\dot{\omega}}_{fu} = -\frac{\bar{\rho}}{\tau_{mix}} \min \left[C_R \widetilde{Y}_{fu}, C_R \frac{\widetilde{Y}_{ox}}{w}, C_R' \frac{\widetilde{Y}_{pr}}{1+w} \right]$$

where C_R and C_R' are model constants.

The model can also be used with finite rate kinetics by taking into account the Arrhenius kinetic rate expression while computing the minimum as follows.

$$\widetilde{\dot{\omega}}_{fu} = -\min \left[\frac{\bar{\rho}}{\tau_{mix}} C_R \widetilde{Y}_{fu}, \frac{\bar{\rho}}{\tau_{mix}} C_R \frac{\widetilde{Y}_{ox}}{w}, \frac{\bar{\rho}}{\tau_{mix}} C_R' \frac{\widetilde{Y}_{pr}}{1+w}, -\widetilde{\dot{\omega}}_{fu,kinetics} \right]$$

There are certain limitations to the applicability of the eddy break-up model. The model cannot be used when the local turbulent and diffusion time scales are very small under which circumstances the minor species example CO, H_2 will be over-predicted due to equilibrium assumption. Sometimes due to vigorous turbulence the reactions may extinct due to very less residence time, and the reaction rate is zero. This phenomenon is know as local extinction and can be accounted by detailed chemistry models example EDC discussed further in the next section. Further since turbulent mixing time τ_{mix} is used to calculate the mean reaction rate, predicting correct turbulence time scales is a prerequisite for using the eddy break-up model.

2.5.2 Eddy Dissipation Concept

The eddy dissipation concept(EDC) model of Ertesvag and Magnussen (2000) is a detailed chemistry model. The EDC model is a modified version of the eddy break-up model which is a good approximation when the chemical kinetics is faster than the overall fine structure mixing . Unlike the eddy break-up model which assumes that turbulent flames are mixing controlled the EDC model is also applicable when the chemical kinetics has a dominating influence [22].

It is known from the turbulence energy cascade theory [11] that the dissipation of turbulence kinetic energy into heat takes place in the fine structures where the smallest eddies are dissipated due to the work done by molecular forces. These small structures have characteristic dimensions of the order of the Kolmogorov scales. The EDC model assumes that chemical reactions occur in these fine structures, where reactants are mixed at molecular scale and react at sufficiently high temperature. The state of these fine structure regions are calculated using either of the following two approaches

- The fast chemistry approach which considers equilibrium of the fine structure regions.
- The detailed chemistry approach where the fine structure regions are treated as well-stirred reactors.

A comprehensive review of the EDC model can be found in Magnussen [22]. Without giving the detailed derivations the model may be summarized as follows.

In the EDC model the mass fraction occupied by the fine structures is defined as

$$\gamma^* = 4.6 \left(\frac{\nu \tilde{\epsilon}}{\tilde{k}^2} \right)^{\frac{1}{2}}$$

where 4.6 is a model constant. The fraction of the fine structure regions which may react is assumed proportional to the ratio between the local concentration of reacted fuel and the total quantity of the fuel that might react.

$$\chi = \frac{\frac{\tilde{Y}_{pr}}{(1+w)}}{\tilde{Y}_{min} + \frac{\tilde{Y}_{pr}}{(1+w)}}$$

where \widetilde{Y}_{pr} is the product mass fraction and $\widetilde{Y}_{min} = \min(\widetilde{Y}_{fu}, \frac{\widetilde{Y}_{ox}}{s})$ is calculated similar to the eddy break-up model.

The reaction rate for a specie s can then be expressed as

$$\widetilde{\omega}_s = -\frac{\bar{\rho}}{\tau_{mix}} \left(\frac{\chi}{1 - \gamma^* \chi} \right) \left(\widetilde{Y}_s - \widetilde{Y}_s^* \right) \quad s = 1, 2 \dots m \quad (2.21)$$

The reaction rate depends on the mean mass fraction \widetilde{Y}_s for a specie s which is known from solving the individual transport equation for the specie and the mass fraction inside the fine structures Y_s^* that must be computed using either the fast chemistry or the detailed chemistry approach.

The reaction rate for the fuel using the fast chemistry approach is determined using the same ideas as the eddy break-up model and is given by

$$\widetilde{\omega}_s = -\frac{\bar{\rho}}{\tau_{mix}} \left(\frac{\chi}{1 - \gamma^* \chi} \right) \min \left(\widetilde{Y}_{fu}, \frac{\widetilde{Y}_{ox}}{s} \right) \quad (2.22)$$

The reaction rate using the detailed chemistry approach is calculated considering the fine structure region as constant-pressure and adiabatic homogeneous reactor. The mass fraction inside the fine structures Y_s^* is obtained by solving a set of ordinary differential equations describing a perfectly stirred reactor (PSR). When detailed chemistry approach is used the reacting fraction of the fine structure χ is set to unity [1].

Chapter 3

Radiative Heat Transfer

3.1 Introduction

Radiation is often an important heat transfer mode in combustion systems along with diffusion and convective heat transfer due to the bulk fluid motion. To illustrate the role of thermal radiation we re-state the governing enthalpy equation 2.14 for laminar combusting flows.

$$\underbrace{\frac{\partial}{\partial t}(\rho h)}_{\text{Transient term}} + \underbrace{\frac{\partial}{\partial x_j}(\rho u_j h)}_{\text{Convection term}} = \underbrace{\frac{\partial}{\partial x_j} \left[\frac{\mu}{Pr_h} \frac{\partial h}{\partial x_j} + \mu \left(\frac{1}{Sc_s} - \frac{1}{Pr_h} \right) \sum_s h_s \frac{\partial Y_s}{\partial x_j} \right]}_{\text{Diffusion term}} + \frac{\partial p}{\partial t} + \underbrace{S_{rad}}_{\text{Radiation term}}$$

The internal distribution of heat sources and sinks due to chemical reactions and the transport of heat by means of diffusion and convection determine the enthalpy distribution due to fluid flow. The third mechanism of heat transfer which acts as a source in the enthalpy equation is thermal radiation caused by energy emission in the form of electromagnetic waves. Radiative energy sources emit a broad-band spectral distribution with maximum energy content at a wavelength determined by the source temperature. All matter with a temperature greater than absolute zero emits thermal radiation in a wide range of wavelengths. The total amount of radiation of all wavelengths increases steeply as the temperature rises and grows as the fourth power of the absolute temperature, as expressed by the Stefan–Boltzmann law [24].

Radiative heat transfer is often neglected in most engineering problems where temperatures are moderate and heat transfer is convection dominated. However in combustion problems where there is usually vigorous convection but the chemical reactions generate operating temperatures that are sufficiently high for radiative fluxes to be of similar order of magnitude [9]. The radiative heat transfer is therefore indispensable for predicting the correct temperatures in furnaces which typically operate in the range 1500-3000K.

In this chapter we introduce the radiative transfer equation (RTE) and its connection with the source term S_{rad} in the energy equation 2.14. This chapter describes one of the most basic model, the P1 approximation model for radiative heat transfer. It offers some insight into the various assumptions for gaseous combustion in industrial furnaces, drawing heavily from the reviews by Coelho, Mancini and Roekaerts [25].

In the following section we introduce the equation describing the transport of radiative energy and its connection with the source term S_{rad} . This term is expressed as the divergence of the radiative heat flux vector derived from the radiative transfer equation (RTE).

3.2 Radiative Transfer Equation

Radiative transfer in a participating medium is mathematically described by the radiative transfer equation (RTE) which describes the rate of change of the spectral radiation intensity of a radiation beam travelling in the medium and propagating along a certain direction. The radiative transfer equation [26] in its most general form is expressed as

$$\frac{dI_\lambda(\mathbf{r}, \mathbf{s})}{ds} = \underbrace{-\kappa_\lambda I_\lambda(\mathbf{r}, \mathbf{s})}_{\text{absorption term}} + \underbrace{\kappa_\lambda I_{b\lambda}(\mathbf{r})}_{\text{emission term}} \underbrace{-\sigma_{s\lambda} I_\lambda(\mathbf{r}, \mathbf{s}) + \frac{\sigma_{s\lambda}}{4\pi} \int_{4\pi} I_\lambda(\mathbf{r}, \mathbf{s}^*) \Omega(\mathbf{s}^*, \mathbf{s}) d\Omega^*}_{\text{scattering term}} \quad (3.1)$$

where I_λ is the spectral radiation intensity at point \mathbf{r} that propagates along direction \mathbf{s} , s is the coordinate along that direction, κ and σ_s are the absorption and scattering coefficients of the medium, respectively, and $\Omega(\mathbf{s}^*, \mathbf{s})$ is the scattering phase function. Subscripts b and λ denote blackbody and wavelength, respectively.

The spectral radiance of a body, I_λ , describes the amount of energy it gives off as radiation of different wavelengths. It is measured in terms of the power emitted per unit area of the body, per unit solid angle that the radiation is measured over, per unit wavelength. Planck showed that the spectral radiance of a blackbody at absolute temperature T is given by

$$I_{b\lambda}(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}.$$

where k_B is the Boltzmann constant, h the Planck constant, and c the speed of light in the medium, whether material or vacuum. It states that the emitted radiation power increases with increasing temperature and that the maximum of the radiated spectrum is shifted towards shorter wavelengths with rising temperature. The total amount of radiation of all wavelengths can be calculated as

$$I_b = \int_0^\infty I_{b\lambda} d\lambda = \frac{\sigma}{\pi} T^4$$

where $\sigma = 5.67 \times 10^{-8} [W/m^2 K^4]$ is the Stefan-Boltzmann constant.

The different terms in the radiative heat transfer equation are due to the interaction of the radiation beam with the participating media [25]. The term on the left represents the rate of change of the spectral radiation intensity of a radiation beam due to the emission, absorption and, scattering (if solid particles are present e.g soot particles) by the participating medium. The first term on the right side of eq. (3.1) represents the attenuation of the spectral radiation intensity within the control volume due to absorption of radiative energy in the medium. The rate of change of spectral radiation intensity due to absorption is directly proportional to the spectral radiation intensity. The proportionality coefficient is the spectral absorption coefficient of the medium, $\kappa_\lambda [m^{-1}]$ which may be interpreted as the inverse of the average distance travelled by a photon until it is absorbed by the medium. This is a property of the medium that depends on its temperature, total pressure and molar fractions of the absorbing species, as well as on the wavelength. The second term on the right side of eq. (3.1) represents the increase of radiative energy within the control volume due to radiative emissions from the medium. It is directly proportional to the blackbody spectral radiation intensity at the temperature of the medium within the control volume. The proportionality coefficient is again the absorption coefficient of the medium. Finally the last two terms in the RTE are due to scattering of radiation within the control volume to directions different from \mathbf{s} . However in combustion systems when the fuel is gas, scattering is negligible [25] and therefore scattering will not be considered any further in this chapter. Additionally the radiative properties of the medium will be assumed to be independent of the spectral variable(λ) for simplicity. Such mediums are called grey and the spectral intensity and total intensity coincide for grey mediums.

The radiative transfer equation describes the radiation intensity $I([W/m^2])$ in a particular direction at a particular wavelength. However we are interested in deriving the transport equation for the total radiation intensity(G) and the divergence of the radiative heat flux vector(\mathbf{q}) which appears as a source or sink term in the enthalpy equation 2.14. The total radiation intensity(G) can be obtained by integrating the spectral radiation intensity I_λ over all possible directions and across all possible wavelengths.

$$G = \int_0^\infty \int_{4\pi} I_\lambda d\Omega \quad (3.2)$$

The spectral radiative flux(\mathbf{q}) is defined as

$$\mathbf{q} = \int_0^\infty \int_{4\pi} I_\lambda \mathbf{s} d\Omega \quad (3.3)$$

An equation for the divergence of the spectral radiative flux(\mathbf{q}) can be obtained by integrating the RTE over all directions spanning a solid angle of 4π and across all possible wavelengths [26].

$$-\nabla \cdot \mathbf{q} = \kappa G - 4\kappa\sigma T^4 \quad (3.4)$$

Here we can observe that the total radiative heat loss is negligible at low temperatures due to the small value of σ . However, the radiative heat loss plays a significant role at very high temperatures (example an object at the temperature of the filament in an incandescent light bulb—roughly 3000 K, or 10 times room temperature—radiates 10,000 times as much energy per unit area than an object at room temperature.).

The divergence of the total radiative heat flux vector at an arbitrary point in space, given by eq. (3.4), represents the amount of energy per unit time and per unit volume that is gained or lost at that point as a result of the global radiative heat exchange. Since scattering does not cause any local change of energy, but only a redistribution of radiative energy among the directions therefore the terms due to scattering are not present in the source term for radiation. Further the absorption coefficient of the medium is independent of the spectral variable(λ) and therefore $\kappa_\lambda = \kappa$ for grey mediums. Therefore, the source term in the enthalpy equation accounting for radiative heat losses is given by:

$$S_{rad} = -\nabla \cdot \mathbf{q} = \underbrace{\kappa G}_{\text{heat gain}} - \underbrace{4\kappa\sigma T^4}_{\text{heat loss}} \quad (3.5)$$

However, the incident total radiation intensity(G) is not known a priori and therefore, it must be calculated by formulating a transport equation for G . There are many solution methods that have been developed over the years to formulate the radiative heat transport equation. In the present work we consider the simplest method called the spherical harmonics method wherein the RTE is transformed into a set of simultaneous partial differential equations. This is accomplished by first expressing the radiation intensity field as a series expansion in terms of directional dependent spherical harmonics and then truncating the series to a few terms(usually one or three terms). The former known as the P1 approximation yields just one partial differential equation and the later yields 16 simultaneous partial differential equations. We restrict ourselves to P1 approximation because of its reasonable accuracy at less computational costs [25].

3.3 P1 Approximation for Radiative Heat Transfer Problems

In the P1 approximation the RTE for a grey non-scattering medium is reduced to the following equation [26]

$$q = -\frac{1}{3\kappa_\lambda} \nabla G \quad (3.6)$$

$$\text{i.e. } \nabla \cdot \left(\frac{1}{3\kappa_\lambda} \nabla G \right) = \kappa G - 4\kappa\sigma T^4 \quad (3.7)$$

subject to the following Marshak boundary condition for a diffuse boundary.

$$q_w = -\frac{\epsilon_w}{2(2 - \epsilon_w)}(4\sigma T_w^4 - G_w) \quad (3.8)$$

$$\text{i.e. } \frac{-1}{3\kappa_\lambda} \mathbf{n} \cdot \nabla G = \frac{\epsilon_w}{2(2 - \epsilon_w)}(4\sigma T_w^4 - G_w) \quad (3.9)$$

where T_w is the calculated wall temperature and G_w is the incident radiation intensity at the wall. ϵ_w is the wall emissivity, which is taken to be 1 for open boundaries, including the inlet and outlet sections of the combustion chamber which closely resemble a black body. At symmetry planes and perfectly reflecting boundaries, the emissivity is taken to be 0 which corresponds to zero radiative heat flux. Mathematically the transport equation for the total incident radiation intensity (G) resembles a diffusion equation with a spatially varying source term that depends on G itself. The P1 approximation is usually only accurate when the radiation intensity field is nearly isotropic [25] under which conditions the divergence of the radiative heat flux reduces to a diffusion equation with a spatially varying source term as a function of temperature only. This is known as optically thick approximation or diffusion approximation and constitutes the so called Rosseland model [27] for radiative heat transport given by.

$$\nabla \cdot q = -\nabla \cdot \left(\frac{16\sigma T^3}{3\kappa} \nabla T \right) \quad (3.10)$$

3.3.1 Limitations of P1 Approximation

The P1 approximation is usually only accurate when the radiation intensity field is nearly isotropic which is true for optically thick media [25]. But for gaseous combustion the optically thick assumption is not suitable for pure molecular gases which are transparent or optically thin over large spectral regions. Therefore, the results of the P1 approximation though satisfactory are not very accurate when compared to the more comprehensive Discrete Ordinate Method [28] and Discrete Transfer Method [29].

The computationally expensive Discrete Ordinate Method [25] (fvDOM model in OpenFOAM) can be used as an alternative to P1 radiation model when more accurate temperature predictions are desired. The fvDOM model solves the RTE equation for a finite number of discrete solid angles, each associated with a vector direction \vec{s} in a participating media. The fineness of the angular discretization is a user input. The Discrete Ordinate Method solves for as many transport equations as there are directions \vec{s} . The computational cost is determined by the total number of solid angles which is equal to $4 \cdot n_{\text{Phi}} \cdot n_{\text{Theta}}$ where n_{Phi} denotes the azimuthal angles in $\text{PI}/2$ on X-Y. (from Y to X) and n_{Theta} denotes the polar angles in PI (from Z to X-Y plane). The fvDOM model is the method of choice for combustion problems where an accurate calculation of the temperature field is desired for correctly predicting the pollutant NO concentrations as illustrated in chapter 5.

Chapter 4

External Wall Heat Transfer

4.1 Introduction

To solve the system of coupled partial differential equations 2.11-2.14, 3.6 we need to prescribe boundary conditions for the following solution variables that appear in the equations i.e. velocity(u_i), pressure(p), enthalpy(h), species mass fraction(Y_s) and incident radiation intensity(G). In this chapter we discuss the appropriate boundary conditions for enthalpy(h).

Since the enthalpy(h) is related to the temperature(T) from 2.16 the boundary conditions for enthalpy(h) can be realized by imposing boundary conditions on the temperature or heat flux. The most commonly used boundary condition is Dirichlet boundary condition for temperature prescribed at the furnace walls. However, an exposed hot surface of an industrial furnace needs to be insulated to reduce the heat loss to the surroundings. Therefore, the temperature at the furnace wall is not known a priori and is calculated as part of the solution procedure for the flow, velocity, temperature distribution and species concentration in the interior of the furnace.

In most applications, heat conduction in a solid needs to be considered along with heat convection in the adjacent fluid. The term conjugate heat transfer is used to describe processes which involve variations of temperature within solids and fluids, due to thermal interaction between them. A typical example is the heating or cooling of a solid wall due to the surrounding air. Figure 4.1 shows the variation of the temperature across an infinite 2D plane wall due to surrounding fluids on either side maintained at different temperatures.

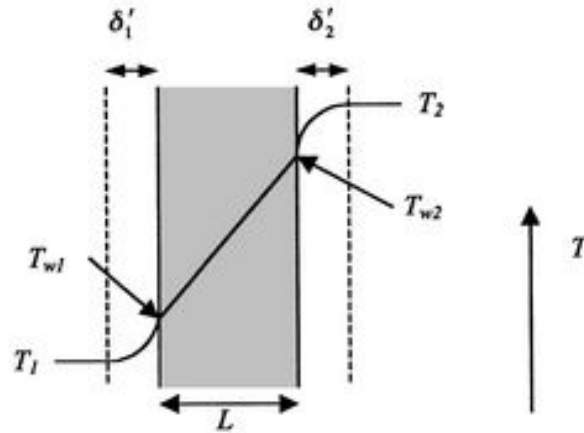


Figure 4.1: 1D heat transfer across a plane wall

The thermal interaction between the fluid and solid body at microscopic scales ensures that the temperature field and the heat flux are continuous at the fluid/solid interface. However, the temperature field can rapidly vary in a fluid in motion. Close to the solid, the fluid temperature is close to the solid temperature, and far from the interface, the fluid temperature is close to the inlet or ambient fluid temperature. The distance where the fluid temperature varies from the solid temperature to the fluid bulk temperature is called the thermal boundary layer. The thermal boundary layer size and the momentum boundary layer relative size is reflected by the Prandtl number. For the Prandtl number to equal 1, thermal and momentum boundary layer thicknesses need to be the same. A thicker momentum layer would result in a Prandtl number larger than 1. Conversely, a Prandtl number smaller than 1 would indicate that the momentum boundary layer is thinner than the thermal boundary layer. The Prandtl number for air at atmospheric pressure and at 20°C is 0.7. That is because for air, the momentum and thermal boundary layer have similar size, while the momentum boundary layer is slightly thinner than the thermal boundary layer. For water at 20°C, the Prandtl number is about 7. So, in water, the temperature changes close to a wall are sharper than the velocity change [30].

The term “Conjugate” refers to the coupling between the fluid and the solid region as a result of thermal interactions at microscopic level. One way to realize conjugation is by iterating between the equations describing the two types of heat transfer in the solid and the adjacent fluid until the temperature field and the heat flux are continuous at the fluid/solid interface. Therefore, to accurately predict the temperature distribution inside the furnace the heat loss from the hot flue gases in the interior to the furnace walls and the cooling of the furnace walls due to ambient air needs to be formulated as a conjugate heat transfer problem. The temperature at the interior of the furnace walls thus can be obtained by solving the corresponding conjugate heat transfer problem.

4.2 1D Approximation for External Wall Heat Transfer

As stated earlier a way to realize conjugation is by iterating between the equations describing the heat transfer in the solid region and the adjacent fluid region until the temperature field and the heat flux are continuous at the fluid/solid interface. This however is computationally very expensive given the fact that the heat transfer in the fluid region is coupled with the more complex species transport and chemical reactions along with the flow. A crude approximation to model external wall heat transfer is to consider a 1D heat flux boundary condition in the normal direction(out of the furnace) for all cells on the domain boundary. This would mean that the heat transfer in the furnace lining could only be in the direction normal to the surface which is a good assumption when the thickness of the wall and the distribution of heat in the adjacent fluid is uniform throughout the interior of the furnace. Under these assumptions the heat loss through the insulating furnace walls into the ambient air can be obtained by calculating the total heat flux due to thermal conduction in the solid and due to natural convective heat transfer into the ambient air [13]. The external heat flux boundary condition at the furnace walls can be computed as.

$$\begin{aligned} q &= h_1(T_{w1} - T_1) = \frac{1}{R}(T_{w2} - T_{w1}) = h_2(T_2 - T_{w2}) \\ \Rightarrow q &= \frac{(T_1 - T_{w2})}{\frac{1}{h_1} + R} \end{aligned}$$

where q is the thermal current and R is the total wall resistance. The total wall resistance can be calculated for multiple insulation layers of thicknesses $L_1, L_2.. L_i$ with respective thermal conductivity's $k_1, k_2.. k_i$ as

$$R = \sum_i \frac{L_i}{k_i}$$

The convective heat transfer coefficient of the ambient air h_1 can be obtained from correlations for natural convection for external flows around horizontal cylinder [31], [32]. These empirical correlations are presented for their particular geometry and flow conditions. As the fluid properties are temperature dependent, they are evaluated at the film temperature T_f , which is the average of the surface T_{w1} and the surrounding bulk temperature T_1 . The fluid-side heat transfer coefficient h_2 is computed based on the local flow-field conditions, turbulence level, temperature, and velocity profiles [13].

Chapter 5

NO_x Formation

5.1 Introduction

The species NO and NO_2 are collectively known as NO_x . NO_x along with SO_x , heavy metal compounds and soot particles are the main pollutants formed during combustion. NO_x is of particular concern because of its impact on the formation of ozone and is directly responsible for acid rain and formation of smog [15]. Therefore, NO_x prediction in any combustion equipment is of utmost importance.

NO_x primarily results from two sources [9].

- Molecular N_2 present in atmospheric air.
- Fuel-bound nitrogen present in the fuels such as heavy oil and coal.

NO_x concentrations generated in combustion systems are generally low. As a result, NO_x chemistry has negligible influence on the predicted flow field, temperature, and major combustion product concentrations [13]. Because of the 1-way coupling between flow and NO_x (i.e. flow influences NO_x distribution but not the other way) the NO_x concentrations can be obtained by solving passive scalar transport equations for NO_x mass fractions. Therefore, the most efficient way to use the NO_x model is as a post processor to the main combustion calculation.

5.2 Sources of NO Formation

Among the NO_x components nitric oxide (NO) is the primary nitrogen oxide emitted from most combustion sources. Formation of NO during combustion processes can be classified into three types based on the source of formation [9].

- **Thermal NO**

Thermal NO is formed at very high temperatures, usually above 1800K, and is a result of the direct oxidation of the diatomic nitrogen found in combustion air. Thermal

NO is the most produced form of NO_x created during combustion. It is a function of the temperature and the residence time of the nitrogen at that temperature, the higher the temperature of the flame the higher the formation of thermal NO. It has been observed that thermal NO production doubles for every 100K temperature increase when the flame temperature is about 2100K as shown in 5.1. It is therefore necessary to predict the temperature distribution accurately for accurate NO_x prediction. The reaction mechanism for thermal NO formation is well known as the extended Zeldovich mechanism discussed in detail in the chapter.

- **Prompt NO**

Prompt NO is produced by high-speed reactions of the atmospheric nitrogen with radicals such as C, CH, and CH₂ fragments derived from fuel in oxygen deficient regions of the flame. Prompt NO is formed in the earliest stage of combustion long before the time required to form thermal NO. It results in the formation of fixed species of nitrogen such as NH, HCN, H₂CN and CN which can oxidize to NO.

It is known that during combustion of hydrocarbon fuels, prompt NO formation rate can exceed that produced from direct oxidation of nitrogen molecules(i.e thermal NO) [13]. The formation of Prompt NO is proportional to the number of carbon atoms present per unit volume and is independent of the parent hydrocarbon. Prompt NO can be formed in a significant quantity in some combustion environments, such as in low-temperature, fuel-rich conditions and where residence times are short e.x. surface burners, staged combustion systems and gas turbines. At present the prompt NO contribution to total NO from stationary combustors including power plants, industrial combustion plants is generally very low, hence it is usually considered only when the most stringent emission targets are to be met.

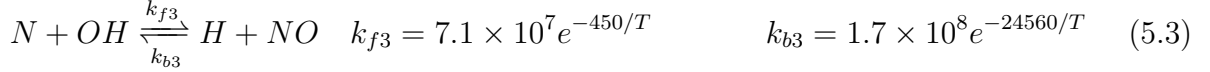
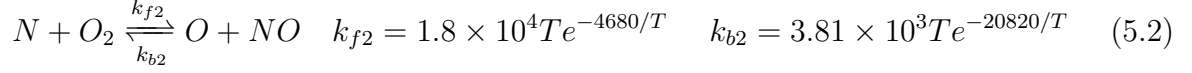
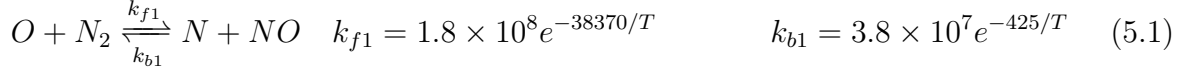
- **Fuel NO**

Fuel NO is produced by oxidation of nitrogen contained in the fuel. In general, a fuel with lower nitrogen content typically produces less total NO than a fuel with higher nitrogen content. Therefore, it is of interest only when the fuel into consideration contains nitrogen (example heavy fuel oils, solid fossil fuels, coal etc).

5.2.1 Thermal NO Formation

Thermal NO reactions are much slower than other combustion reactions and therefore its formation takes too long to achieve equilibrium and hence cannot be calculated using equilibrium chemistry models. The formation of thermal NO is determined by a set of highly temperature-dependent chemical reactions known as the extended Zeldovich mechanism which is often the method of choice for post processing thermal NO formation. The principal reactions governing the formation of thermal NO from molecular nitrogen are [33]

:



The net rate of formation of NO via the reaction eqs. (5.1) to (5.3) is given by

$$\frac{d[NO]}{dt} = k_{f1}[O][N_2] + k_{f2}[N][O_2] + k_{f3}[N][OH] - k_{b1}[NO][N] - k_{b2}[NO][O] - k_{b3}[NO][H] \quad (5.4)$$

where k_{f1} , k_{f2} , k_{f3} are the forward reaction rate constants and k_{b1} , k_{b2} , k_{b3} are the backward reaction rate constants in units of $m^3/mol - s$.

The Quasi-Steady Assumption for N atoms [13]

The high activation energy of forward reaction 5.1, which breaks the strong N_2 triple bond is the rate-limiting step of the extended Zeldovich mechanism. It is because of this high activation energy that the NO production takes place after the combustion reactions are complete and after significant heat is generated in the system. However, the activation energy for oxidation of N atoms is small and when there is sufficient oxygen, as in a fuel-lean flame, the rate of consumption of free nitrogen atoms becomes equal to the rate of its formation, and therefore a quasi steady state is reached [13]. The quasi steady state assumption is valid for most combustion cases, except in extremely fuel-rich combustion conditions. This approach gives the best prediction for fuel lean air mixtures under high flame temperature conditions [34]. Using the quasi steady state assumption the formation rate becomes

$$\frac{d[NO]}{dt} = 2 k_{f1} [O] [N_2] \frac{\left(1 - \frac{k_{b1} k_{b2} [NO]^2}{k_{f1} k_{f2} [N_2] [O_2]}\right)}{\left(1 + \frac{k_{b1} [NO]}{k_{f2} [O_2] + k_{f3} [OH]}\right)} \quad (mol/m^3 - s) \quad (5.5)$$

From the limiting rate described by 5.1 it is evident that thermal NO formation rate is very sensitive to temperature. The thermal NO production rate doubles for every 100K temperature increase when the flame temperature is about 2100K as depicted in fig. 5.1. Further it can be observed that thermal NO formation rate increases for higher O atom concentration and is independent of fuel type.

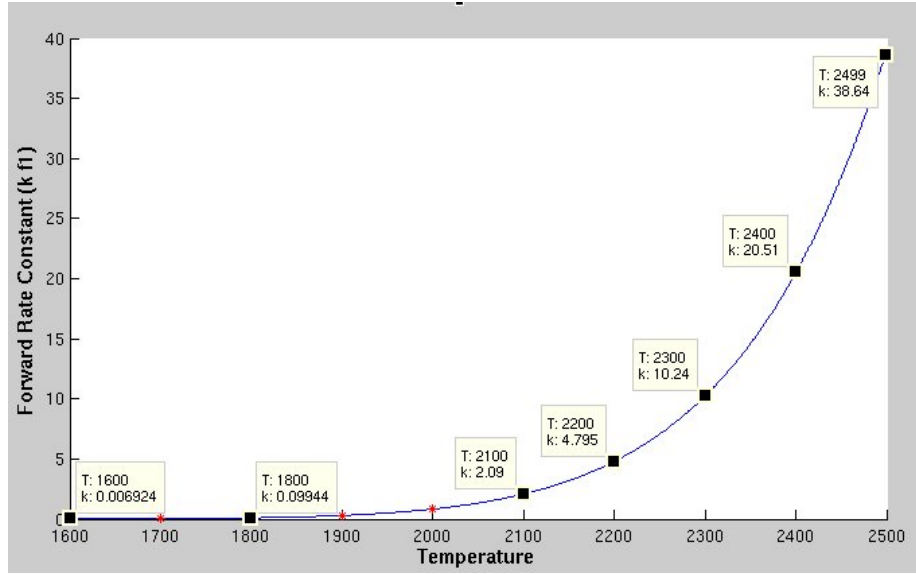


Figure 5.1: Thermal NO production rate sensitivity with temperature

The rate of NO formation governed by equation 5.5 is a function of T, global species O_2 and N_2 concentration and concentration of O atoms and OH radicals. The global species concentration is known from the combustion simulation using simplified reaction mechanism, however O and OH radical concentrations are not known a priori and have to be determined. It is known that the kinetics of thermal NO formation is much slower than the main hydrocarbon oxidation rate, and most of the thermal NO is formed after completion of combustion. Therefore, the thermal NO formation process can often be decoupled from the main combustion reaction mechanism and the NO formation rate can be calculated by assuming equilibrium concentrations of O atoms and OH radical as illustrated by ANSYS Fluent user guide [13].

Determining O radical concentration

- Equilibrium Chemistry Approach

$$[O] = 3.97 * 10^5 T^{-1/2} e^{-31090/T} [O_2]^{1/2} mol/m^3 \quad (5.6)$$

- Partial Equilibrium Approach

The O atoms concentration has been observed to be more abundant than their equilibrium levels. The level of thermal NO can be under predicted by as much as 28% when considering Equilibrium Approach [13]. The more accurate partial equilibrium approach accounts for third-body reactions in the O_2 dissociation recombination process and generally leads to a higher partial O atoms concentration.

$$[O] = 36.64 T^{1/2} e^{-27123/T} [O_2]^{1/2} mol/m^3 \quad (5.7)$$

Determining OH radical concentration

- Exclusion of OH Approach

The third body reaction in the extended Zeldovich mechanism contributes to the formation of NO at near-stoichiometric conditions or in fuel-rich mixtures and is negligible for fuel lean conditions which leads to the following assumption $k_{f2} [O_2] \gg k_{f3} [OH]$.

- Partial Equilibrium Approach

$$[OH] = 2.129 * 10^2 T^{-0.57} e^{-4595/T} [O]^{1/2} [H_2O]^{1/2} mol/m^3 \quad (5.8)$$

5.3 Calculating NO Mass Fraction

The NO species transport equation needs to be solved for obtaining the NO mass fraction (Y_{NO}) distribution due to convection, diffusion and volume sources. The governing convection diffusion equation for NO transport is given by,

$$\underbrace{\frac{\partial \rho Y_{NO}}{\partial t}}_{\text{Transient term}} + \underbrace{\nabla \cdot (\rho \vec{u} Y_{NO})}_{\text{Convection term}} = \underbrace{\nabla \cdot (\rho D_{eff} \nabla Y_{NO})}_{\text{Diffusion term}} + \underbrace{S_{Y_{NO}}}_{\text{Source term}}$$

where $S_{Y_{NO}}$ is the source term accounting for production of NO calculated from the rate of change of concentration of NO in equation eq. (5.5).

$$S_{Y_{NO}} = M_{NO} \frac{d[NO]}{dt} Kg/m^3 - s \quad (5.9)$$

The NO concentration can be obtained from the NO mass fraction using the relation

$$[NO] = \frac{p Y_{NO}}{RT}$$

Since the thermal NO production is very sensitive to temperature distribution as illustrated earlier, an accurate temperature distribution is a prerequisite for reasonable NO prediction.

Chapter 6

Implementation in OpenFOAM

6.1 Introduction

This chapter gives a brief overview of the OpenFOAM toolbox used for CFD simulations in the present thesis. In section 6.3 the new solver NOxFoam for simulating the transport of a passive scalar is described to familiarize the reader of the structure of the OpenFOAM toolbox. The more complex solvers simpleFoam for turbulent isothermal flows and furnaceFoam for turbulent reacting flows are discussed only in brief in section 6.4 and section 6.5 respectively.

6.2 OpenFOAM

OpenFOAM or “Open Field Operation and Manipulation” is a CFD toolbox written in C++, used primarily to create executables, known as applications. The applications fall into two categories 1) solvers and 2) utilities. Solvers are each designed to solve a specific problem in continuum mechanics; and utilities, are designed to perform tasks that involve data manipulation. The OpenFOAM distribution contains numerous solvers and utilities capable of solving anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetism [35]. It also includes tools for meshing, notably snappyHexMesh - a parallelised mesh generation tool for complex CAD geometries, utilities for post-processing and data manipulation. The overall structure of OpenFOAM is shown in fig. 6.1. Almost everything (including mesh generation and post-processing) runs in parallel as standard, enabling users to take full advantage of the computer hardware at their disposal.

There are many CFD packages available both commercial and open source for CFD applications. Some of the commercial packages have dedicated solvers for turbulent combustion for instance, STAR CCM+ and ANSYS Fluent. Although these packages are being extensively used and proven to be reliable, they are extremely expensive with huge license costs. This research was therefore, carried out using the free open source technology Open-

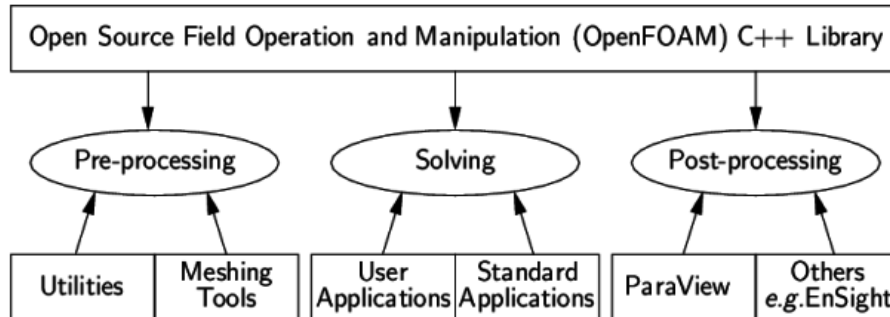


Figure 6.1: OpenFOAM applications: solvers and utilities [35]

FOAM. Some of the reasons besides no license fee which motivated us to use OpenFOAM are.

- By being open, OpenFOAM offers users complete freedom to customize and extend its existing functionality. Contrary to many open source packages it follows a highly modular code design in which collections of functionality (e.g. numerical methods, physical models, . . . etc) are each compiled into their own shared library. Executable applications are then created that are simply linked to the existing library functionality.
- OpenFOAM has a very large and active user community across the globe including researchers from academic institutions and industry [36]. This ensures frequent bug reports and revised versions unlike any other open source package. In the Netherlands OpenFOAM is widely being used at many university's and research institutes example Nuclear Research and consultancy Group(NRG) [37]. It is also becoming very popular in industry for instance Tata Steel Research & Development [38] and Dyna Flow Research Group are actively using OpenFOAM.
- It has the highest level academic references and detailed validation and verification studies for several well-known in the literature combustion test cases [1, 2, 3].

As every coin has two sides, OpenFOAM has the following disadvantages when compared to its commercial counterparts.

- There is no official documentation present describing the OpenFOAM code layout, its solvers and utility's. However, besides referring to the source code of a solver directly the user can refer to the C++ documentation generated using doxygen, thesis reports such as this and training sessions at OpenFOAM annual workshops.
- It does not provide an integrated graphical user interface and therefore for setting up an OpenFOAM case the user has to write his own dictionary files listing the physical models, boundary conditions, discretisation schemes, solver tolerances etc. Recently

there has been new developments such as the HELYX Open Source Graphical User Interface for OpenFOAM. However, they do not offer support for the complex solvers such as reactingFoam for combustion. Further, OpenFOAM foundation currently distributes OpenFOAM for Linux operating systems only and the user therefore needs to have an adequate experience with UNIX like systems to get familiar with OpenFOAM.

- OpenFOAM does not have its own post-processing utilities for data visualization, monitoring residuals, probing data at selective points in domain etc. OpenFOAM is often used in conjunction with ParaView for data visualization, PyFoam for monitoring residuals, generating line plots, generating case reports etc. An important external library used with OpenFOAM is swak4Foam. This library offers a number of utilities (for instance funkySetFields to set fields using expression), boundary conditions (groovyBC to specify arbitrary boundary conditions based on expressions) and function objects that allow doing many things that would otherwise require programming.

6.2.1 OpenFOAM Structure

OpenFOAM employs cell-centered, colocated finite volume discretisation to solve systems of partial differential equations on any structured or unstructured mesh. The model equations are first linearized and then solved in a segregated manner by using the classical SIMPLE/PISO or hybrid PIMPLE algorithms. In the following subsection the OpenFOAM structure is illustrated using the case setup for turbulent reacting flows. The details of the finite volume discretisation methodology used by OpenFOAM are further explained in section 6.3.

A problem to be solved is represented in OpenFOAM as a case directory with individual sub-directories for mesh, boundary conditions and solver settings. The basic directory structure for a OpenFOAM case, that contains the minimum set of files required to run an application, is shown in fig. 6.2. In the present work OpenFOAM release 2.3.0 was used for all implementations and subsequent simulations.

The constant directory

This is the first place where a user begins setting up his/her case in OpenFOAM. The domain is discretised into a finite number of control volumes using either native OpenFOAM utilities blockMesh, snappyHexMesh or external programs such as Salome or Gambit. OpenFOAM supports unstructured meshes of cells of any shape, cells can have any number of faces and faces can have any number of edges. Meshes generated by any of the major mesh generators can be converted to OpenFOAM format using conversion tools provided by OpenFOAM. The mesh is contained in the sub-directory polyMesh.

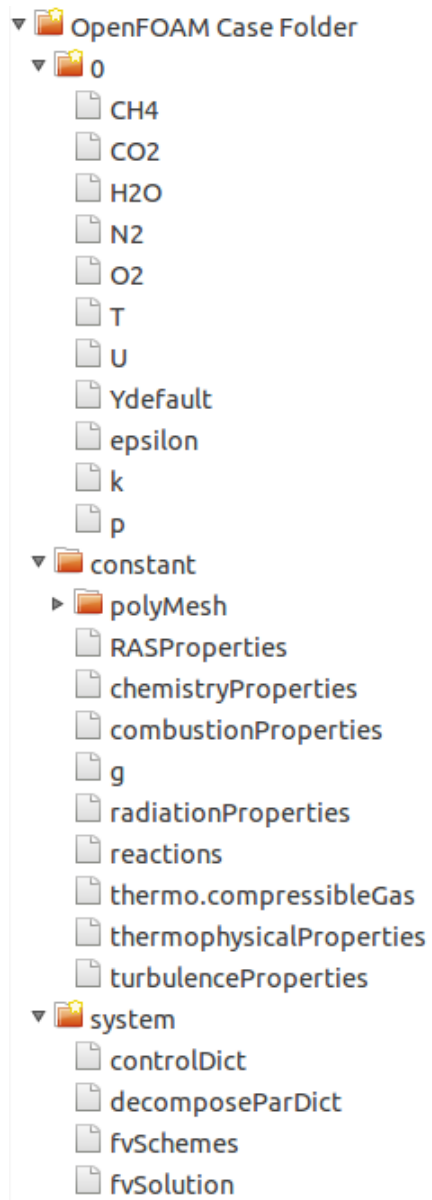


Figure 6.2: OpenFOAM case directory structure

Besides the mesh the constant directory is the place where the user specifies all the physical properties for the application concerned. OpenFOAM uses dictionaries as the most common means of specifying data. A dictionary is an entity that contains data entries that can be retrieved by the I/O by means of keywords. For example if the user intends to simulate turbulent combustion, the following dictionary files are required in the constant directory.

- g - lists the standard acceleration due to gravity.
- turbulenceProperties - lists the turbulence model.
- RASProperties - lists the turbulence model constants.
- chemistryProperties - lists the initial chemical time step, ODE solver, and solver properties to be used.
- combustionProperties - lists the turbulence chemistry interaction model and model constants.
- radiationProperties - lists the radiation model and model constants.
- thermophysicalProperties - lists the equation of state, transport law (viscosity as a function of temperature), representing energy form (enthalpy, internal energy) etc.
- reactions - lists the detailed reaction mechanism in native OpenFOAM format (or CHEMKIN format).
- thermo.compressibleGas - lists the thermophysical data for the participating chemical species.

The system directory

This is the place where the user specifies parameters associated with the solution procedure itself. It contains at least the following three files: controlDict where run control parameters are set including start/end time, time step and parameters for data output; fvSchemes where discretisation schemes for time derivatives, gradient terms, divergence terms, laplacian terms, interpolation schemes, surface normal gradient schemes etc used in the solution process may be selected at run-time; and, fvSolution where the equation solvers, pre-conditioners, tolerances and other algorithm controls are set for the run. An important thing to note is that the choice of the schemes and solution parameters can be changed at run time without stopping the solution process. This enables the user for instance to change from the more robust but diffusive first order discretisation schemes to more accurate second order schemes after the first few thousand iterations when the solution is stable and accuracy is now desired rather than robustness[39].

The time directories

They contain individual files of data for particular fields. The data can be: either, initial values and boundary conditions that the user must specify to define the problem;

or, results written to file by OpenFOAM. The name of each time directory is based on the simulated time at which the data is written. In fig. 6.2 the dictionary files for velocity field (U), pressure (p), temperature (T), chemical species mass fractions (CH4, CO2, H2O, N2, O2, Ydefault), turbulent kinetic energy (k) and dissipation (epsilon) are contained in the 0 time directory where the solution process begins. The same files also prescribe the boundary conditions for the corresponding variables.

6.3 Solver for Thermal NO Prediction - NOxFoam

A passive scalar is a diffusive contaminant in a fluid flow that is present in such low concentration that it has no dynamical effect on the fluid motion itself [40]. For example contamination of the atmospheric air due to plumes of black smoke emerging from an industry. The smoke from the chimneys gets advected by the velocity field but does not contribute to its dynamics. NO_x concentrations generated in combustion systems are generally low. As a result, NO_x chemistry has negligible influence on the predicted flow field, temperature, and major combustion product concentrations [13]. Therefore, NO_x concentrations can be obtained by solving passive scalar transport equations for NO_x mass fractions.

The passive scalar transport of NO mass fraction is implemented in OpenFOAM using the existing scalarTransportFoam solver. The scalarTransportFoam solver is upgraded to take into consideration the spatial variation of density and mass diffusivity. The source term accounting for NO formation according to the extended Zeldovich mechanism eq. (5.5) is added to the passive scalar transport equation for NO. The governing equation for transport of NO mass fraction introduced in section 5.3 is re-stated in eq. (6.1) where ϕ denotes the NO mass fraction.

$$\underbrace{\frac{\partial \rho \phi}{\partial t}}_{\text{Transient term}} + \underbrace{\nabla \cdot (\rho \vec{u} \phi)}_{\text{Convection term}} - \underbrace{\nabla \cdot (\rho D_{eff} \nabla \phi)}_{\text{Diffusion term}} = \underbrace{S_\phi}_{\text{Source term}} \quad (6.1)$$

One distinguishing feature of OpenFOAM is the way it represents the physics of the problem to be solved. Its syntax for tensor operations and partial differential equations closely resembles the equations being solved. For example eq. (6.1) is represented in OpenFOAM terminology as shown in fig. 6.3.


```

solve
(
    fvm::ddt(rhoAvg,NO)
  + fvm::div(fvc::interpolate(rhoAvg)*phi, NO)
  - fvm::laplacian(rhoDTeff, NO)
  ==
    sourceThermalNO
);

```

Figure 6.3: OpenFOAM syntax

OpenFOAM employs collocated finite volume discretisation to solve systems of partial differential equations. In this method the domain is first decomposed into a finite number of disjoint control volumes (CV) of general polyhedral shape as shown in fig. 6.4. The unknown variables are located at the centroid of each control volume thereby giving the collocated variable arrangement. The control volumes are bounded by a set of flat faces and each face is shared with only one neighboring control volume.

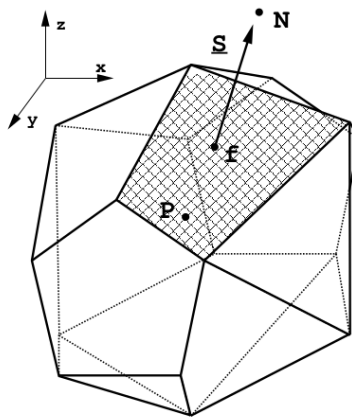


Figure 3.1: Control volume.

Figure 6.4: Finite control volume [41]

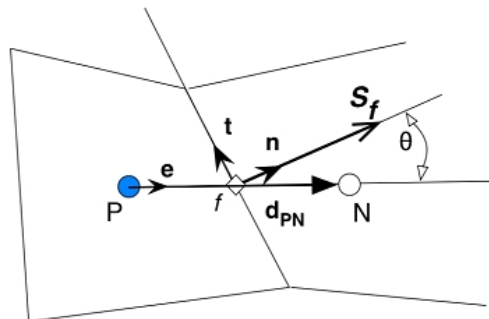


Figure 6.5: A general non-orthogonal control volume [42]

The integral form of the governing partial differential equation is discretised over each control volume conserving mass and momentum, a characteristic of the finite volume discretisation methodology. The integral form of eq. (6.1) is given by.

$$\int_t^{t+\delta t} \left[\frac{\partial}{\partial t} \int_{\Omega} \rho \phi dV + \int_{\Omega} \nabla \cdot (\rho \vec{u} \phi) dV - \int_{\Omega} \nabla \cdot (\rho D_{eff} \nabla \phi) dV \right] dt = \int_t^{t+\delta t} \int_{\Omega} S_{\phi}(\phi) dV dt \quad (6.2)$$

Discretisation of convection term

For a general control volume shown in fig. 6.5, applying Gauss divergence theorem the discretization of the convection term can be written in the following form.

$$\int_{\Omega} \nabla \cdot (\rho \vec{u} \phi) dV = \int_{\partial \Omega} (\phi \rho \vec{u}) \cdot d\vec{S} = \sum_{faces} \phi_f (\rho \vec{u})_f \cdot \vec{S}_f = \sum_{faces} \phi_f F_f$$

where \vec{S}_f denotes the area vector of the face f and $F_f = (\rho \vec{u})_f \cdot \vec{S}_f$ is the mass flux through the face. The flux is calculated from the interpolated values of ρ and \vec{u} from the cell center to the face center. The face value of the variable ϕ is similarly obtained from the neighbouring cell values i.e. ϕ_P and ϕ_N using suitable interpolation scheme specified in the fvSchemes file. In the present simulations the linearUpwind scheme was used which is a first/second order bounded scheme derived from upwind and linear schemes. The choice of the scheme is a compromise between accuracy and boundedness [41].

Since ρ is a volume field, it needs to be interpolated to obtain its value on the face. This is achieved in the code using `fv::interpolate(rhoAvg)`, where `fv` (finite volume calculus) is a C++ class for creating new fields and `interpolate` is the `fv` class method. The divergence is then computed using `div` method of the `fvm` class.

Discretisation of diffusion term

Applying Gauss divergence theorem the discretization of the diffusion term can be written in the following form.

$$\int_{\Omega} \nabla \cdot (\rho D_{eff} \nabla \phi) dV = \int_{\partial \Omega} (\rho D_{eff} \nabla \phi) \cdot d\vec{S} = \sum_{faces} (\rho D_{eff})_f (\nabla \phi)_f \cdot \vec{S}_f$$

The default approach that is used for approximating the diffusive flux D_f [41] at the

face f is

$$D_f = (\rho D_{eff})_f \nabla \phi_f \cdot \vec{S}_f = (\rho D_{eff})_f \left[(\phi_N - \phi_P) \left| \frac{\vec{S}_f}{d_{PN}} \right| \right] \quad (6.3)$$

where $(\rho D_{eff})_f$ is obtained from the neighbouring cell values using interpolation scheme specified in the fvSchemes file. This results in an approximation for the gradient in the direction of the vector connecting the two cell centers. This approach is not very good when the non-orthogonality of the grid is high. A common procedure used to correct for the mis-alignment between the line joining the cell centers and the face normal is to use orthogonality corrections.

The term $\nabla \phi_f \cdot \vec{S}_f$ can be expressed as the sum of orthogonal contribution and non-orthogonal correction.

$$\nabla \phi_f \cdot \vec{S} = \underbrace{\nabla \phi_f \cdot \vec{S}_d}_{\text{orthogonal contribution}} + \underbrace{\nabla \phi_f \cdot \vec{\Delta}}_{\text{correction}} \quad (6.4)$$

where $\vec{S} = \vec{S}_d + \vec{\Delta}$ and \vec{S}_d is chosen to be parallel with \vec{d} . The orthogonal contribution is then determined according to eq. (6.3) and the correction term is calculated by interpolating the gradients at the neighbouring cell centers P and N as follows

$$(\nabla \phi)_f = \alpha (\nabla \phi)_P + (1 - \alpha) (\nabla \phi)_N$$

where α is a weighing factor for interpolation and

$$(\nabla \phi)_P = \frac{1}{V_P} \sum_{\text{faces}} \phi_f \vec{S}$$

There are different ways to decompose \vec{S} and the most common approach used is the orthogonal correction approach where

$$\Delta = \frac{\vec{d}}{|\vec{d}|} |\vec{S}| \quad (6.5)$$

The reader is referred to [41], [42], [43] for more information on the different approaches used for non-orthogonality corrections. In the present simulations linear scheme was used for interpolation of $(\rho D_{eff})_f$ on the face and orthogonal correction scheme was used for decomposition of the surface normal gradient \vec{S} . This is achieved in the code using the laplacian method of the fvm (finite volume method) class.

Discretisation of the source term

The source term can be a non-linear function of the transported scalar ϕ as in the case of NO transport where the formation of NO is a function of the NO concentration given in eq. (5.5). Therefore, the source term is first linearised before computing the volume integral.

$$S_\phi(\phi) = S_u + S_p\phi$$

$$\int_{\Omega} S_\phi(\phi) dV = S_u(\phi)V_p + S_p(\phi)V_p\phi \quad (6.6)$$

Discretisation in time

The discretisation in space results into the following semi-discretised form of the transport eq. (6.2).

$$LHS = \int_t^{t+\delta t} \left[\left(\frac{\partial \rho \phi}{\partial t} \right)_P V_P + \sum_{faces} \phi_f F_f - \sum_{faces} (\rho D_{eff})_f (\nabla \phi)_f \cdot \vec{S}_f \right] dt \quad (6.7)$$

$$RHS = \int_t^{t+\delta t} S_u(\phi)V_p + S_p(\phi)V_p\phi dt \quad (6.8)$$

Using an implicit time discretisation scheme (example Euler implicit or Crank-Nicolson), the time derivative can be approximated as

$$\left(\frac{\partial \rho \phi}{\partial t} \right)_P = \frac{\rho_P^n \phi_P^n - \rho_P^0 \phi_P^0}{\Delta t} \quad (6.9)$$

This is achieved in the code using ddt method of the fvm class.

It can be observed that all the terms of the equations were discretized implicitly, using the fvm (finite volume method) class which is used for creating the matrix representation of an operator.

The discretised equation for each control volume P can be assembled in the form of an algebraic system (for details see Jasak [41]) of the form.

$$a_p \phi_p^n + \sum_N a_N \phi_N^n = S_P \quad (6.10)$$

The assembled system is then solved for the unknown ϕ for the next time step using the method solve() in the code. In the present work the first order, bounded, implicit Euler scheme is used for time discretisation. The large sparse linear system resulting from the assembly of the discretised equations for each control volume are solved using preconditioned bi-conjugate gradient (PBiCG) with diagonal incomplete LU (DILU) as the preconditioner.

Calculation of S_ϕ , ρ and ρD_{eff} for NO transport

The implementation details of the passive scalar transport equation for NO mass fraction was presented above. However, it requires the calculation of the source term S_ϕ , density ρ and mass diffusivity D_{eff} . The source term for NO transport given by eq. (5.5) is re-stated here as follows

$$S_{Y_{NO}} = M_{NO} \quad 2 k_{f1} [O] [N_2] \frac{\left(1 - \frac{k_{b1} k_{b2} [NO]^2}{k_{f1} k_{f2} [N_2] [O_2]}\right)}{\left(1 + \frac{k_{b1} [NO]}{k_{f2} [O_2] + k_{f3} [OH]}\right)} \quad (mol/m^3 - s)$$

The following quantities are known from the main combustion calculations

- Temperature(T) distribution.
- The velocity field (\vec{u}).
- The mass fraction Y_s of the participating chemical species and pressure(p) distribution.
- Turbulent viscosity μ_t .

From the molecular weights and the available mass fractions of the stable species the weighted density is computed as follows $\rho = M_{avg} \frac{P}{RT}$, $M_{avg} = \sum_s f_s M_s$ where M_s is the molar mass of specie s. The effective molecular diffusivity is given by

$$\rho D_{eff} = \frac{\mu_o}{Sc} + \frac{\mu_t}{Sc_t}$$

where Sc and Sc_t are the Schmidt number and turbulent Schmidt number respectively and μ_o is the laminar viscosity computed using Sutherland law for Transport

$$\mu_o = A_s \frac{T^{3/2}}{T + T_s}$$

where $A_s = 1.67212 \times 10^{-6}$, $T_s = 170.672$. The average Schmidt number for NO in air of 0.85 is used for calculations. The turbulent Schmidt number of 0.7 is used for comparison with ANSYS Fluent.

6.4 Solver for Turbulent Isothermal Flows - simpleFoam

Simulating turbulent reacting flows is no trivial task because of the strong coupling between flow and chemistry. Initializing both flow and chemistry at the same time can cause instabilities and therefore turbulence and velocity fields from an isothermal flow simulation are often used as an initial guess for simulating turbulent reacting flows.

The simpleFoam solver in OpenFOAM is a steady-state solver for incompressible, turbulent flows. The solver is used for simulating turbulent isothermal (Cold) flows in chapter 9. It is based upon the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm [44] for pressure velocity coupling. The application of the SIMPLE algorithm in the finite volume framework is illustrated in this section without providing the implementation details of the solver. For implementation details the reader is referred to [14] and Weller [45].

The unsteady Reynolds-averaged Navier-Stokes equations were stated earlier in section 2.2.1 (eq. (2.7), eq. (2.8)). When steady state is reached the mean flow is time invariant and therefore, the time derivative term can be dropped to obtain the steady RANS equations.

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \quad (6.11)$$

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial}{\partial x_i} \left[\bar{p} + \frac{2}{3} \rho k \right] + \frac{\partial}{\partial x_j} \left[\nu_{eff} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right], \quad i = 1, 2, 3 \quad (6.12)$$

These can be written in compact vector form as

Continuity equation

$$\nabla \cdot \vec{u} = 0 \quad (6.13)$$

Momentum equation

$$\nabla \cdot (\vec{u}\vec{u}) - \nabla \cdot (\nu_{eff} \nabla \vec{u}) = -\nabla p_{eff} \quad (6.14)$$

where $p_{eff} = \bar{p} + \frac{2}{3} \rho k$ and $\frac{1}{\rho}$ factor is dropped in OpenFOAM in front of the pressure term for constant density flows. Therefore, if the true mean pressure field is sought for the term has to be taken into account. Further it can be observed that the pressure term is not explicitly present in the continuity or momentum equations but only its gradient is present in the momentum equation, therefore the absolute value of pressure is not

important but only the changes of pressure in space are important for incompressible flows.

The solution of the coupled system of equations (6.13, 6.14) is not straightforward because of the following two reasons.

- An explicit equation for the pressure is not available.

A possible solution is to derive an equation for the pressure by taking the divergence of the momentum equation and by substituting it in the continuity equation.

- The momentum equation is non-linear due to the convection term. The discretised form of this equation would be quadratic in velocity and therefore the resulting system of algebraic equations will be non-linear.

A possible solution to the problem is the segregated approach of Patankar, S. V. and Spalding, D.B. (1972) [46], where the partial differential equations are solved one at a time, with the inter-equation coupling treated in the explicit manner. Non-linear equations are first linearised before the discretisation and the non-linear terms are lagged.

The simpleFoam solver in OpenFOAM is based on the SIMPLE algorithm of Patankar and Spalding [47]. The implementation of the algorithm is shown in fig. 6.6 with correspondence to the source code listed in appendix A. For implementation details of the algorithm and the C++ constructs used for representing the operators the reader is referred to [14], [41], [45].

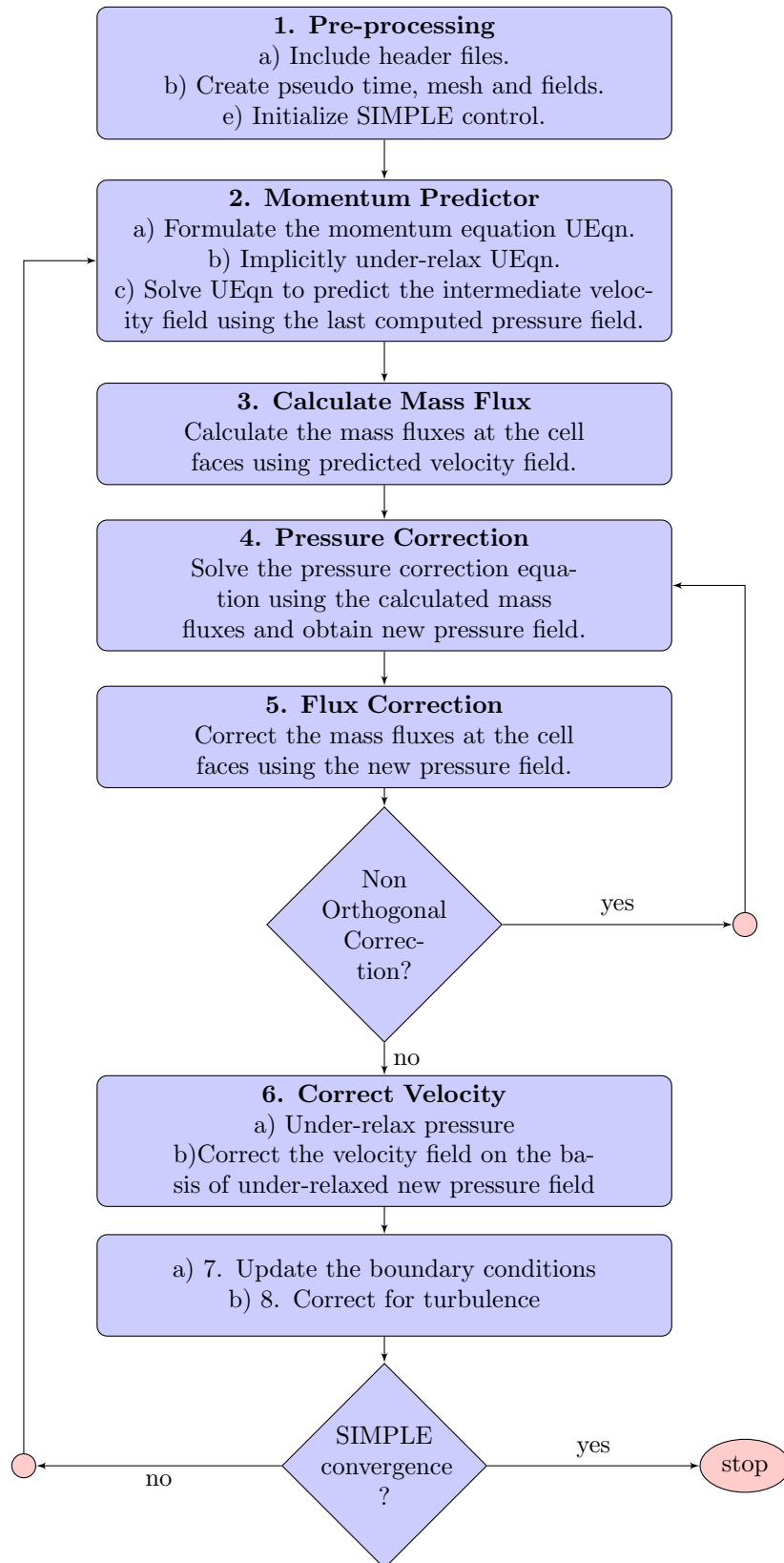


Figure 6.6: Flow diagram of the solver simpleFoam (appendix A) (OpenFOAM 2.3.0.)

6.5 Solver for Turbulent, Reacting Flows + Radiation - furnaceFoam

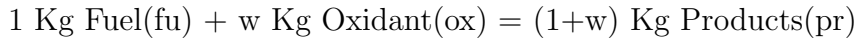
OpenFOAM is capable of simulating non-premixed turbulent combustion of gaseous hydrocarbon fuels. There exist many solvers in OpenFOAM-2.3.0 which can simulate gas combustion including reactingFoam, rhoReactingFoam, fireFoam, XiFoam and PDRFoam. In the present work the solver reactingFoam was used to derive the new solver furnaceFoam that includes radiation modelling. furnaceFoam is used for simulating incompressible turbulent reacting flows in chapter 10.

reactingFoam is a transient solver for non-premixed turbulent combustion of gaseous hydrocarbon fuels. It is an incompressible solver based upon the PIMPLE(merged PISO - SIMPLE [48]) algorithm for pressure velocity coupling. An alternative to reactingFoam is the solver rhoReactingFoam suitable for compressible flows. Since the observed velocity in industrial furnaces is much below the compressibility limit of 0.3 Mach the compressibility effects can be neglected and therefore reactingFoam was considered for our application.

In section 2.5 the two commonly used turbulence chemistry interaction models in industry - the eddy break-up model and the eddy dissipation concept were illustrated. However these models are not available with the solver reactingFoam in OpenFOAM-2.3.0. The turbulence-chemistry interaction models available with reactingFoam are as follows.

infinitelyFastChemistry

Simple infinitely fast chemistry combustion model based on the principle mixed is burnt. It is derived from the singleStepCombustion base class for combustion using singleStepReactingMixture given by



The reaction rate is calculated as

$$\widetilde{\omega}_{fu} = \frac{\bar{\rho}}{C \Delta t} \min \left[\widetilde{Y}_{fu}, \frac{\widetilde{Y}_{ox}}{w} \right] \quad (6.15)$$

where Δt is the flow time step and additional parameter C is used to distribute the heat release rate in time.

diffusion

Simple diffusion-based combustion model based on the principle mixed is burnt. It is also derived from the singleStepCombustion base class. The reaction rate is calculated as

$$\widetilde{\omega}_{fu} = C \mu_{eff} \left(|\nabla \widetilde{Y}_{fu} \cdot \nabla \widetilde{Y}_{ox}| \sqrt{|\widetilde{Y}_{fu}| |\widetilde{Y}_{ox}|} \right) \quad (6.16)$$

Additional parameter C is used to distribute the heat release rate in time.

PaSR

The PaSR model similar to the EDC model section 2.5.2 is a detailed chemistry model. It is derived from the laminar base class for combustion using detailed chemistry. The model calculates a finite rate, based on both turbulence and chemistry time scales.

In the PaSR model [49, 50] a computational cell is divided into a reacting part, and a non-reacting part. The reacting part is treated like a perfectly stirred reactor where all the chemical species present are assumed to be homogeneously mixed and reacted. After reactions have taken place, the species are mixed due to turbulence for the mixing time τ_{mix} , and the resulting concentration gives the final concentration in the entire, partially stirred, cell [51].

The reaction rate is calculated as

$$\tilde{\omega}_s = \frac{\tau_c}{\tau_c + \tau_{mix}} \dot{\omega}_{s,laminar} \quad s = 1, 2 \dots m \quad (6.17)$$

where τ_c is the chemical time scale determined by solving the coupled ODEs governing the chemical kinetics. The turbulence mixing time τ_{mix} is calculated as

$$\tau_{mix} = C_{mix} \sqrt{\frac{\mu_{eff}}{\rho \tilde{\epsilon}}}$$

Depending on mesh resolution, the C_{mix} parameter can be used to scale the turbulence mixing time scale. In the present work the default value of $C_{mix} = 0.1$ was used.

From the available solvers in OpenFOAM-2.3.0 the solver reactingFoam is the most suitable choice for our application. However it does not include radiation modelling which is a pre-requisite for predicting accurate temperature distribution in combustion chambers which operate at high temperatures 1500-3000 K. Therefore, the solver reactingFoam was modified to obtain the new solver furnaceFoam shown in fig. 6.7 that also includes radiation modelling. The modifications essentially include the following steps.

- Including the header files for radiation modelling in the enthalpy equation.
- Including the source term eq. (3.5) in the enthalpy equation accounting for radiative heat losses.
- Including the radiative transfer equation for calculating transport of the incident total radiation intensity (G).
- Including radiation libraries in the make file for linking the new application against the existing radiation libraries during compilation.

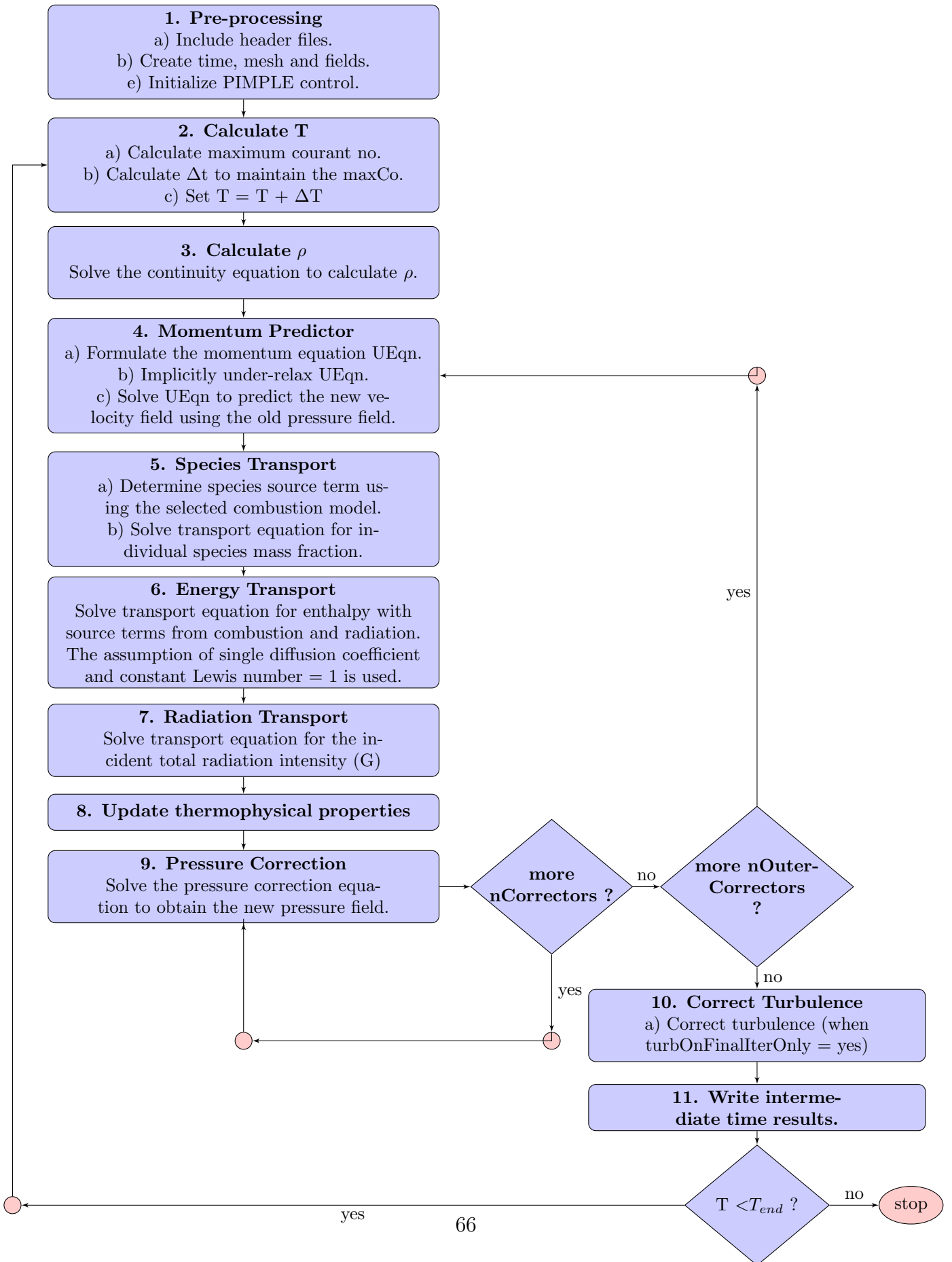


Figure 6.7: Flow diagram of the solver furnaceFoam (appendix B) derived from reactingFoam (OpenFOAM 2.3.0.)

Chapter 7

Sandia Flame D Validation

7.1 Introduction

In this chapter the new solver `furnaceFoam` described in section 6.5 is validated using Sandia National Laboratories Flame D (Sandia Flame D). Sandia Flame D is a turbulent piloted methane jet flame. Sandia Flame D experiment [52] provides a standard and high quality reference data set for validation of turbulent flame calculations. It is one of the most widely used benchmark problem for validating new combustion solvers.

In the present work, the simulation results are compared with the experimental data provided by Barlow and Frank [4]. The standard $k - \epsilon$ model for turbulence and Partially Stirred Reactor(PaSR) model for combustion are used for comparison. The radiation is treated by the P1-approximation model. The simulation results are obtained using simplified reaction mechanisms including the global 1-step reaction mechanism [13], 2-step mechanism of Westbrook and Dryer [17] and 4-step mechanism of Jones and Lindstedt [17]. These are compared with the detailed reaction mechanism GRI-Mech 3.0 [18] along with the experimental data provided by Barlow and Frank.

This chapter is divided into eight sections including the introduction section itself. In section 7.2 the experimental setup of Sandia Flame D, the geometry and the computational domain are described. The boundary and initial conditions for the Sandia Flame D test case are presented in section 7.3. In section 7.4 the different computational meshes used for grid-independence tests and the grid-independent results are presented. The need of the new solver `furnaceFoam` is illustrated in section 7.5 by studying the effect of radiative heat transfer on the predicted temperature distribution in the furnace. In section 7.6 the 1-step, 2-step and 4-step reaction mechanisms for methane combustion are compared. Finally the comparison of the simulation results with the experimental data are presented followed by a summary of the chapter.

7.2 Experimental Setup

The Flame D from the Sandia/TNF workshop is a piloted methane-air diffusion flame [4]. The central main jet consists of a 25/75%(by volume) methane-air mixture. The fuel has been premixed with air in order to minimize the formation of polycyclic aromatic hydrocarbons and soot. The central main jet is surrounded by a pilot jet and a slow coflow of air outside. The hot mixture from the pilot jet besides stabilising the main jet is also responsible for igniting the fuel which is injected from the main jet. The pilot jet is further surrounded by an air-co-flow after the burner exit. The bulk velocities of the main jet, pilot and air-co-flow are $U_\infty = 49.6$ m/s, $U_p = 11.4$ m/s and $U_c = 0.9$ m/s, respectively. The Reynolds number for the main jet is $Re = 2.24 \times 10^4$ based on the nozzle diameter $d = 7.2$ mm and the bulk jet velocity 49.6 m/s. The pilot flame was burning a mixture of C_2H_2 , H_2 , air, CO_2 , and N_2 with an enthalpy and equilibrium composition that is equivalent to a mixture of methane and air at an equivalence ratio of $\phi = 0.77$. The fuel jet, pilot and co-flow compositions are specified in terms of the species mass fractions calculated from the experimental data documented in detail by Barlow and Frank. All of the details regarding this flame are provided and regularly updated in the web site [52]. The experimental data includes Raman/Rayleigh/LIF measurements of F, T, N₂, O₂, CH₄, CO₂, H₂O, H₂, CO, OH, and NO. Results include axial profiles in the flame ($x/d = 5, 10, 15, \dots, 80$) and radial profiles ($x/d = 1, 2, 3, 7.5, 15, 30, 45, 60, 75$).

The fuel nozzle has a diameter $d = 7.2$ mm and is enclosed by a broad pilot nozzle with inner and outer diameters of 7.7 mm and 18.2 mm respectively as shown in fig. 7.1. Pre-inlet pipes for the fuel-jet and pilot are included in the computational domain in order to obtain fully-developed turbulent velocity profiles at the burner exit. The length of the pre-inlet fuel and pilot pipes are approximately $15 \times d$. The axial and radial dimensions of the computational domain after the burner exit were set to $76.5 \times d$ and $20.83 \times d$, respectively as shown in fig. 7.2.

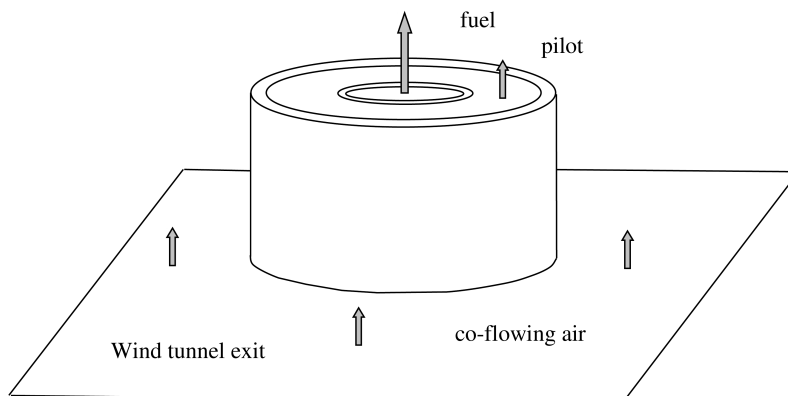


Figure 7.1: Main jet, pilot and coflow [53]

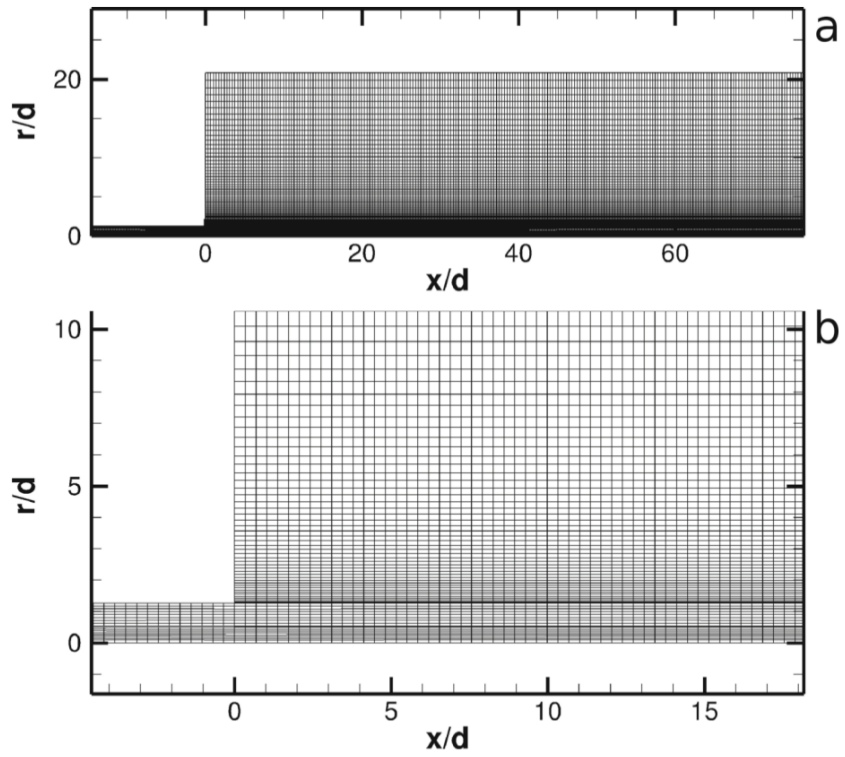


Figure 7.2: a) Computational domain b) mesh near the inlet [3]

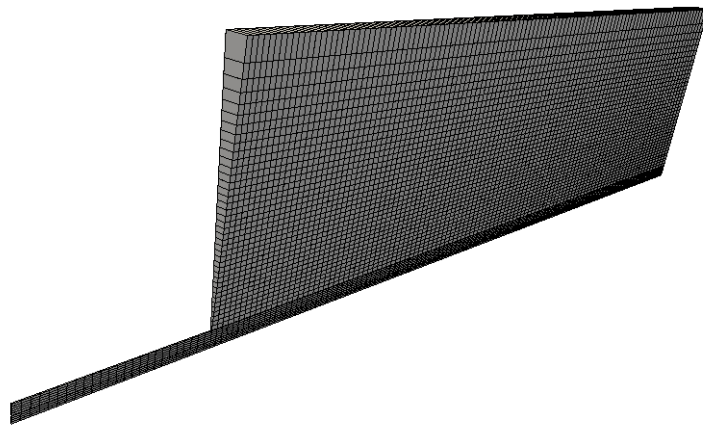


Figure 7.3: Axisymmetric 1-cell thick mesh with wedge angle (angle of cone) = 5° .

7.3 Boundary and Initial Conditions

Dirichlet boundary condition is prescribed at the inlet for all variables except for pressure, that uses a Neumann boundary condition where the normal derivative i.e. gradient is zero. The inflow temperatures are set based on experimental data. No-slip walls are used, together with a zero gradient for other scalar variables. The outlet is located at the top of the computational domain where ambient conditions are prescribed. For the outflow, Neumann boundary conditions are applied for all the scalar variables, except for pressure which is fixed at ambient pressure. For the radiation modelling with P1-method, the Marshaks boundary condition 3.8 is applied for radiative heat flux calculation at the inlets, outlet and walls. The initial turbulent kinetic energy is defined using the turbulence intensity while the initial dissipation rate is defined by the associated mixing length. The standard wall functions for turbulence kinetic energy and dissipation are used for near wall treatment.

Domain	Pressure	Velocity	T(K)	CH_4	O_2	N_2	H_2O	CO_2
main jet	zG	(49.6, 0, 0)	294	0.1561	0.1966	0.6473	0	0
pilot jet	zG	(11.4, 0, 0)	1880	0	0.054	0.742	0.0942	0.1098
air-co-flow	zG	(0.9, 0, 0)	291	0	0.23	0.77	0	0
wall tube	zG	(0, 0, 0)	zG	zG	zG	zG	zG	zG
wall outside	zG	(0.9, 0, 0)	zG	zG	zG	zG	zG	zG
Outlet	1e+5	zG	zG	zG	zG	zG	zG	zG
Internal Volume Field	1e+5	0	291	0	0.23	0.77	0	0

zG - zeroGradient

Velocity expressed in (axial, radial, tangential) components.

mass fraction of species used.

all quantities in SI units.

Table 7.1: Boundary and initial conditions for Sandia Flame D.

7.4 Computational Domain and Grid Independence Tests

The problem is solved assuming symmetry about the center line. The computational domain was designed as a sector of 5° fig. 7.3 with imposed periodic boundary conditions to reduce the computational cost. Since the domain is regular and can be sub-divided into rectangular blocks, a block structured mesh is the most appropriate choice. A 2D block structured mesh is made in Salome, an open source integration platform for numerical simulation. It is then rotated about the center line to generate a sector of 5° .

The simulations are carried out on three different grids with increasing grid density in order to ensure a grid-independent solution. The coarse grid(C1) has 110 and 40 cells along the axial and radial directions, respectively. The total number of cells including the pre-inlet fuel and pilot pipes are 5700 hexahedra and 135 prisms. The 1st refined grid(R1) is derived from C1 by a subsequent grid refinement by factor 2 in both axial and radial directions and comprises of 23070 hexahedra and 270 prisms. The 2nd refined grid(R2) is obtained by considering a uniform grading for the pilot tube(1mm at inlet to 0.5mm at burner), axial(0.5mm at burner to 3mm at outflow) and radial(0.5mm at burner to 1mm outer wall) length from the burner and comprises of 45282 hexahedra and 540 prisms.

The four important parameters that determine the quality of any grid are mesh non-orthogonality, skewness, aspect ratio and smoothness [54] [39] [55]. These four parameters for the grids C1, R1 and R2 are listed in 7.2. Non-orthogonality and Skewness are significant contributors to error in OpenFOAM as illustrated in chapter 8. Since the meshes C1, R1 and R2 are orthogonal and the skewness present is negligible, they are not critical factors in determining the quality of the results for this case using OpenFOAM.

Mesh (#Cells)	Avg Non-Orthogonality	Max Skewness	Max Aspect Ratio
C1 (5835)	0	0.33	27.35
R1 (23340)	0	0.33	26.91
R2 (45822)	0	0.33	12.40

Table 7.2: Mesh quality metrics for C1, R1 and R2

The temperature and the global species (CH_4 , O_2 , CO_2 , H_2O) mass fraction distribution along the central axis of the flame are compared for the grids C1, R1 and R2. The simulation results with grid R1 and R2 are found to be in good agreement with each other. The grid-independent solution is illustrated in fig. 7.4. Therefore, all subsequent comparisons are done using grid R1 except for the detailed reaction mechanism GRI-Mech 3.0 where coarse grid C1 is used due to the very high computational costs table 7.3.

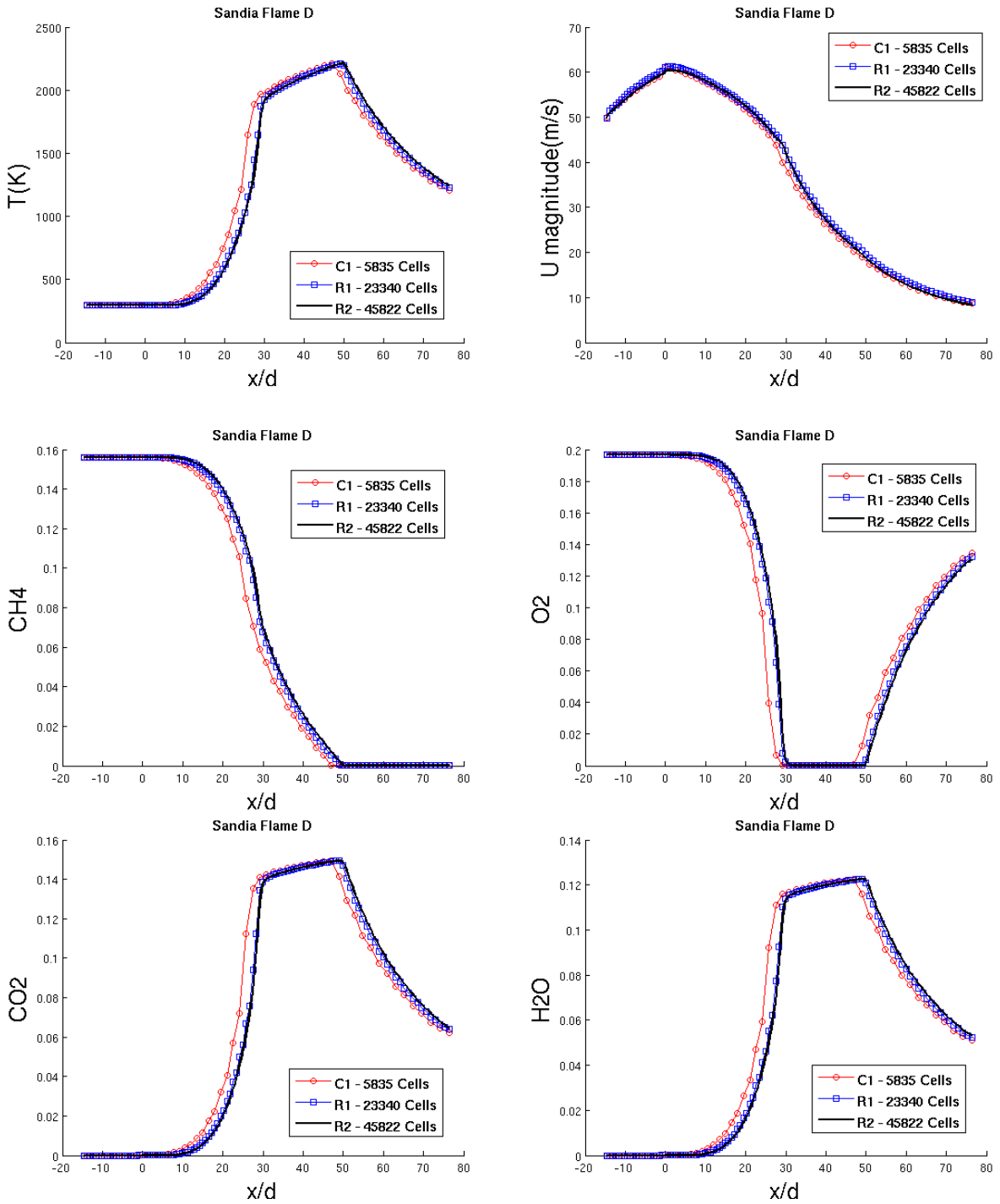


Figure 7.4: Grid-Independence Test ($k-\epsilon$ turbulence model, global 1-step reaction mechanism)

7.5 Effect of Radiative Heat Transfer on the Temperature Distribution

The solver `furnaceFoam` is derived from the existing solver `reactingFoam` by integrating the radiation libraries into the existing solver. The need for this integration is due to the significant contribution of radiative heat transfer in cooling the flame zone, which otherwise leads to over predicted flame temperatures. The effect of radiative cooling can be observed in the temperature plots along the axial and radial positions in the flame as shown in figures 7.5 and 7.6 respectively. The simulation results were obtained using $k-\epsilon$ turbulence model and global 1-step reaction mechanism. Without considering radiative cooling the maximum temperatures are over predicted by 100-150 K. Since determining the maximum temperatures accurately is a pre-requisite for correct thermal NO prediction, including radiative heat losses in the combustion simulations is indispensable. The index in P1-index denotes the number of flow iteration's per solve of the radiative transfer equation(RTE). To minimize the computational costs it would be sufficient to solve the RTE when there are significant temperature changes between the flow iterations. The optimum value of the index will depend on the problem and can be determined by observing the temperature fluctuations between successive iterations in test runs.

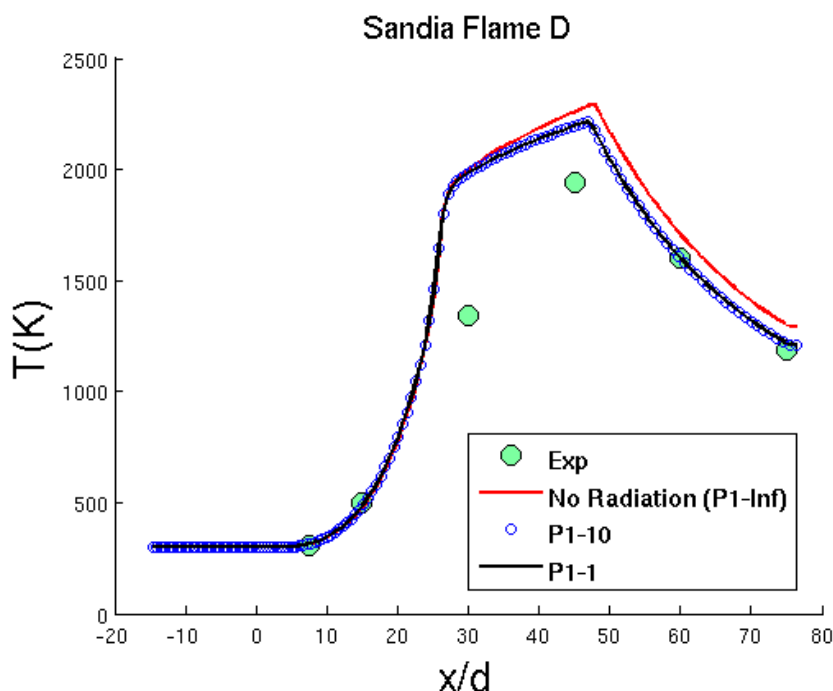


Figure 7.5: Temperature distribution along the central axis at $r/d = 0$.

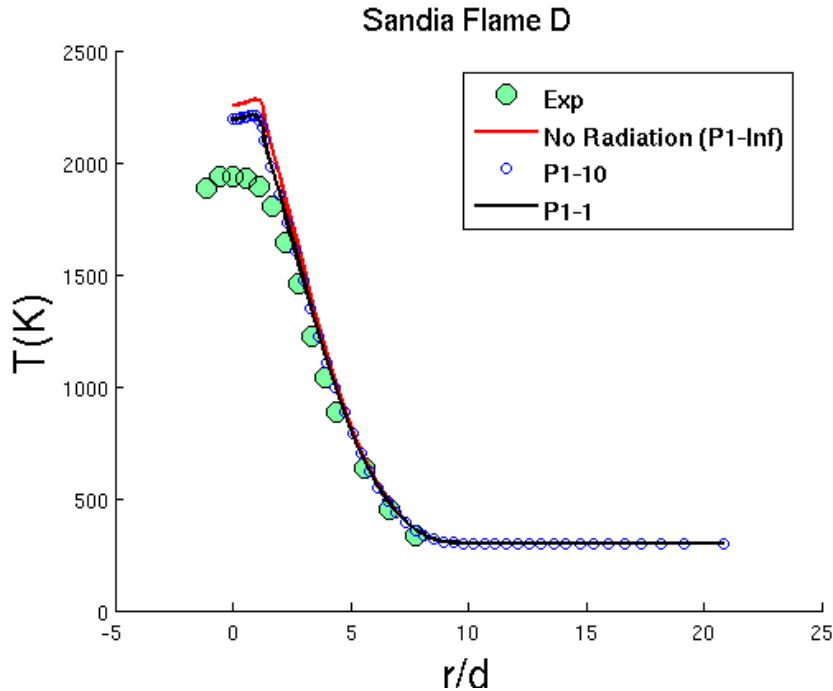
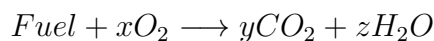


Figure 7.6: Temperature profile in radial direction at $x/d = 45$.

7.6 Different Reaction Mechanisms for Methane Combustion

Simplified reaction mechanisms for methane combustion are compared with the detailed chemical kinetic mechanism GRI-Mech 3.0 [18] which is an optimized mechanism designed to model natural gas combustion, including NO formation. The detailed mechanism contains 325 reactions and 53 species.

The global reaction representing the oxidation of a conventional hydrocarbon fuel is often a convenient way to describe the effects of the many elementary reactions which occur naturally. The reaction in its simplest form can be expressed as



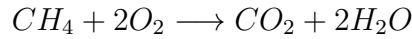
with the rate expression

$$k = AT^\beta e^{-Ea/RT} [Fuel]^a [O_2]^b$$

assumed to be first order in both fuel and oxidizer i.e $a=1$, $b=1$. However, the assumption of first order reaction rate in both the fuel and oxidizer concentration fails to reproduce the correct experimental data and therefore optimization studies are carried out to predict the best values of the concentration exponents a and b for best agreement with experimental data [56].

The simplified reaction mechanisms for methane combustion compared are as follows. The axial and radial profiles for temperature distribution comparing all the 4 approaches with the experimental data is shown in fig. 7.8 and fig. 7.9 respectively. The standard $k - \epsilon$ (ke) model for turbulence and P1 approximation for radiative heat transfer are used for comparison.

- **Single step global reaction mechanism [13]**



Parameters: $A = 6.70 \times 10^{12}$, $E_a = 4.84 \times 10^4$, $[CH_4]^{0.2}$, $[O_2]^{1.3}$

The CHEMKINIII format for the reaction data is shown in figure fig. 7.7.

- **Two-step reaction mechanism of Westbrook and Dryer(1981) [17]**



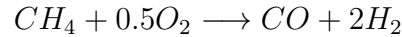
Parameters: $A = 1.59 \times 10^{13}$, $E_a = 4.78 \times 10^4$, $[CH_4]^{0.7}$, $[O_2]^{0.8}$



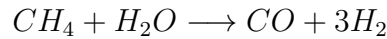
Forward rate parameters: $A = 3.98 \times 10^{14}$, $E_a = 4.07 \times 10^4$, $[CO]^{1.0}$, $[H_2O]^{0.5}$, $[O_2]^{0.25}$

Backward rate parameters: $A = 5 \times 10^8$, $E_a = 4.07 \times 10^4$, $[CO_2]^{1.0}$

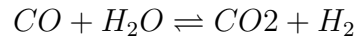
- **Four-step reaction mechanism of Jones and Lindstedt [17]**



Parameters: $A = 7.82 \times 10^{13}$, $E_a = 2.99 \times 10^4$, $[CH_4]^{0.5}$, $[O_2]^{1.25}$



Parameters: $A = 3 \times 10^{11}$, $E_a = 2.99 \times 10^4$, $[CH_4]^{1.0}$, $[O_2]^{1.0}$



Forward rate parameters: $A = 2.75 \times 10^{12}$, $E_a = 2.00 \times 10^4$, $[CO]^{1.0}$, $[H_2O]^{1.0}$

Backward rate parameters: $A = 6.71 \times 10^{13}$, $E_a = 2.72 \times 10^4$, $[CO_2]^{1.0}$, $[H_2]^{1.0}$



Forward rate parameters: $A = 1.79 \times 10^{13}$, $E_a = 3.49 \times 10^4$, $[H_2]^{1.0}$, $[O_2]^{0.5}$

Backward rate parameters: $A = 2.51 \times 10^{14}$, $E_a = 9.51 \times 10^4$, $[H_2O]^{1.0}$

Note: The thermodynamics and reactions data files are provided as input to OpenFOAM in CHEMKINIII format (cm^3 , mole, cal, K). The units of A are $(cm^3/mole)^{r-1}s^{-1}$ and Ea are $cal/mole$. Temperature exponent $\beta = 0$ for all the rate expressions above. The rate expressions for the above reactions are manually converted to CHEMKINIII format, and chemkinToFoam utility is used to check consistency.

chemkinToFoam

The utility chemkinToFoam can be used to convert CHEMKINIII thermodynamics and reaction data files into native OpenFOAM format (m^3 , kmole, Joules, K) for checking consistency of the different reaction mechanism.

$$A(cm^3/mole)^{r-1}s^{-1} \xrightarrow[\text{chemkinToFoam}]{\times 10^{-3(r-1)}} A(m^3/kmole)^{r-1}s^{-1}$$

$$Ea(cal/mole) \xrightarrow[\text{chemkinToFoam}]{\times 4.184/R} Ta(K)$$

where $r = a+b$, $R = 8.314 \text{ J/molK}$ is the universal gas constant and 4.184 is the calories to joules conversion factor.

```

ELEMENTS
H O C N
END
SPECIE
O2 H2O CO2 N2 CH4
END
REACTIONS
CH4 + 2O2 => CO2 + 2H2O 6.70087E12 0 4.84465E+4
      FORD / CH4 0.2 /
      FORD / O2 1.3 /
END

```

```

species
5
(
O2
H2O
CO2
N2
CH4
)
;
reactions
{
un-named-reaction-0
{
type irreversibleArrheniusReaction;
reaction "CH4^0.2 + 2O2^1.3 = CO2 + 2H2O";
A 2.119e+11;
beta 0;
Ta 24377.9;
}
}

```

Figure 7.7: The Global Reaction Mechanism for Methane converted from CHEMKINIII to OpenFOAM format using chemkinToFoam

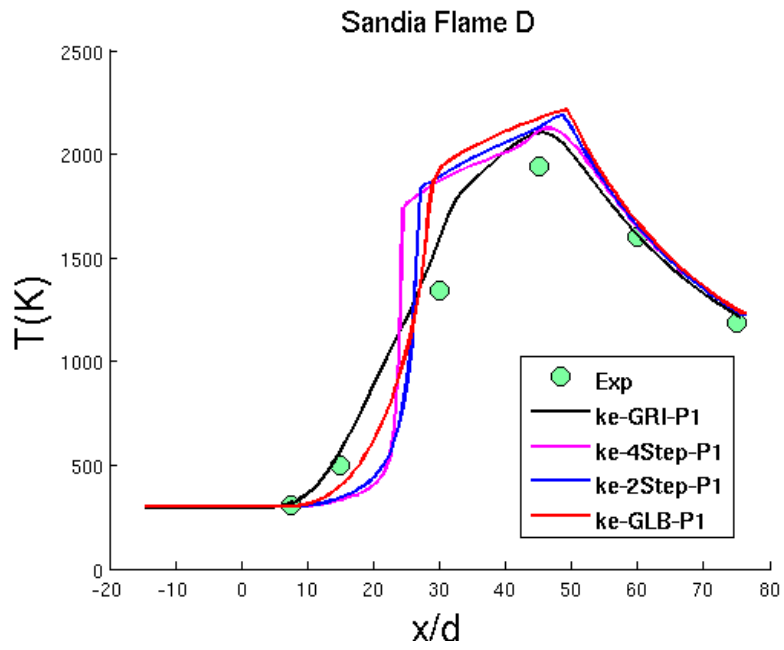


Figure 7.8: Temperature distribution along the central axis at $r/d = 0$.

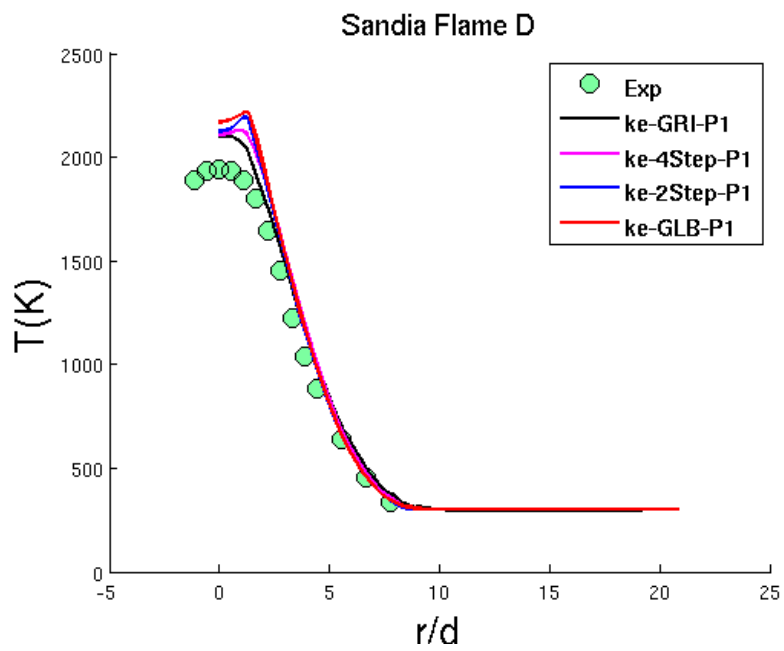


Figure 7.9: Temperature profile in radial direction at $x/d = 45$.

7.7 Comparison of Simulation Results with Experimental Data

A comparison of the predicted flame temperature along the central axis of the flame is shown in fig. 7.8. The maximum temperature along the axis is slightly over predicted by all the different simplified reaction mechanisms. The best agreement is with the detailed chemistry approach as expected. The same is observed for the temperature distribution in the radial direction at a distance of $x = 45 d$ from the burner exit as shown in fig. 7.9.

A comparison of the global species mass fractions along the central axis of the flame is shown in fig. 7.10. The results are in good agreement with the experimental data for the detailed reaction mechanism GRI-Mech 3.0. The small differences in the predicted values by the GRI-Mech 3.0 reaction mechanism is ascribed to the coarse mesh(C1) and the basic models used for turbulence and radiative heat transfer.

The predicted global species mass fractions using the 1-step global reaction mechanism do not agree very well with the experimental data. This is attributed to the fact that the production/consumption of the global species due to the intermediate reactions involving radicals cannot be accounted for in the global 1-step reaction mechanism.

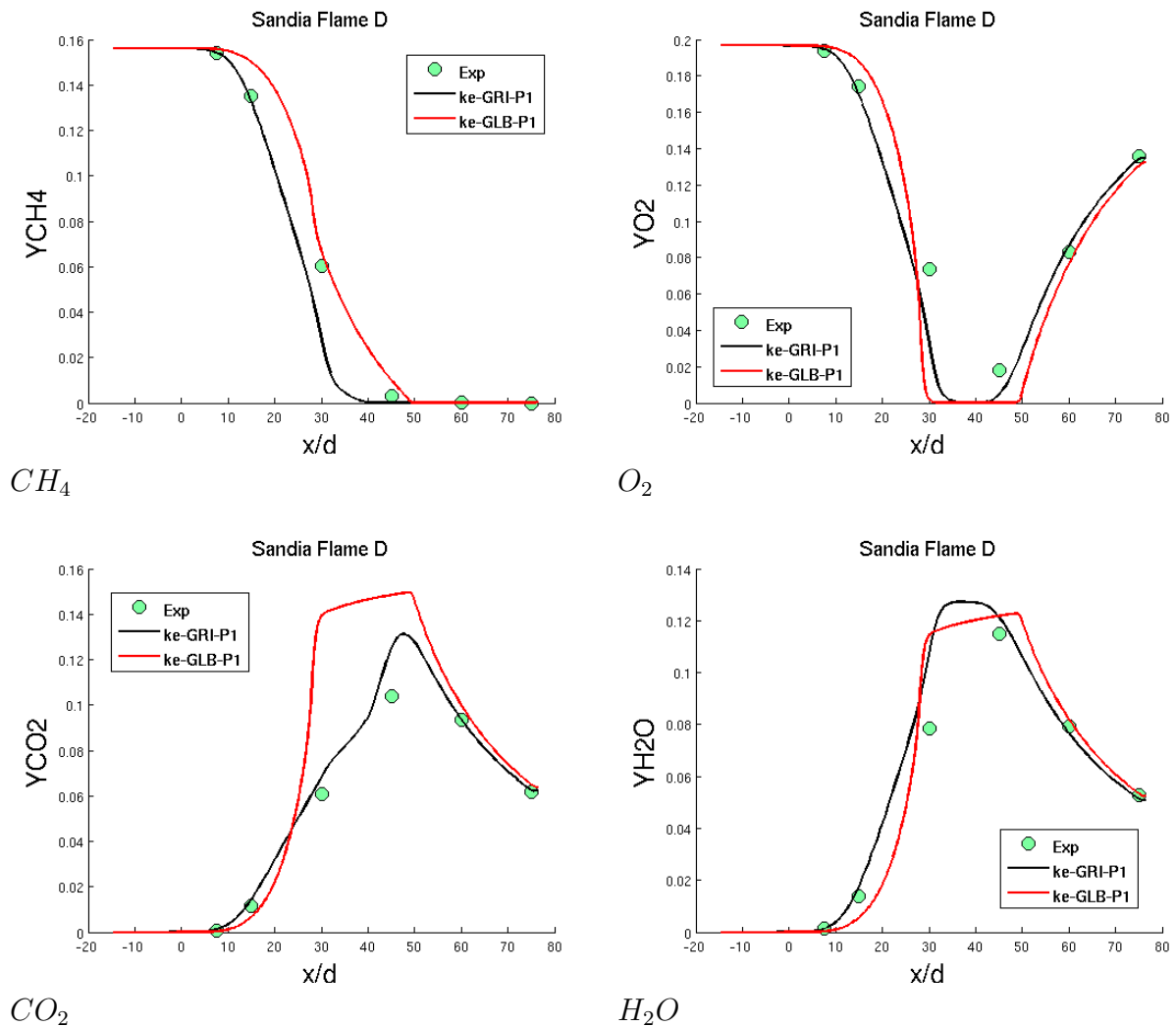


Figure 7.10: Mass fractions along the central axis($r/d=0$) for the global species.

7.8 Computational Time

The reacting flow simulations in OpenFOAM generate a log file containing the residuals of the variables solved for during the simulation and any custom fields defined on patches for example area weighted temperature at the outlet patch (- called monitors in ANSYS Fluent). The raw data dumped in a log file as shown in fig. 7.11 can be used to obtain ANSYS Fluent like plots for the residuals fig. 7.12 and monitors fig. 7.13 using an external utility PyFoam. PyFoam is a open-source python library to control OpenFOAM-runs and manipulate OpenFOAM-data [57].

```

Courant Number mean: 0.0032469517 max: 0.18900026
deltaT = 1.0567557e-05
Time = 0.12

diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
smoothSolver: Solving for O2, Initial residual = 1.2748046e-06, Final residual = 9.3712269e-10, No Iterations 2
smoothSolver: Solving for H2O, Initial residual = 1.3308908e-06, Final residual = 1.0758197e-09, No Iterations 2
smoothSolver: Solving for CO2, Initial residual = 1.3364754e-06, Final residual = 1.0812223e-09, No Iterations 2
smoothSolver: Solving for CH4, Initial residual = 6.7942645e-07, Final residual = 2.6710545e-09, No Iterations 2
smoothSolver: Solving for h, Initial residual = 1.4578732e-06, Final residual = 1.068112e-09, No Iterations 2
min/max(T) = 291, 2230.5729
smoothSolver: Solving for p, Initial residual = 0.0042522561, Final residual = 9.9413768e-09, No Iterations 212
diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
time step continuity errors : sum local = 2.5616952e-08, global = 1.3323349e-08, cumulative = 0.00030945917
smoothSolver: Solving for p, Initial residual = 2.6006363e-06, Final residual = 9.7911915e-09, No Iterations 72
diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
time step continuity errors : sum local = 2.5622486e-08, global = 1.3324664e-08, cumulative = 0.0003094725
smoothSolver: Solving for epsilon, Initial residual = 1.0182587e-06, Final residual = 5.4808205e-09, No Iterations 2
smoothSolver: Solving for k, Initial residual = 2.4373761e-06, Final residual = 5.2233517e-09, No Iterations 2
ExecutionTime = 1902.05 s ClockTime = 2690 s

cellSource volumeTemperature output:
  average(sampledSurface) for T = 797.70819

faceSource faceTemperature output:
  average(outlet) for T = 757.62583

```

Figure 7.11: Snippet of the log file generated by OpenFOAM for the Sandia Flame D simulation - GLB-C1.

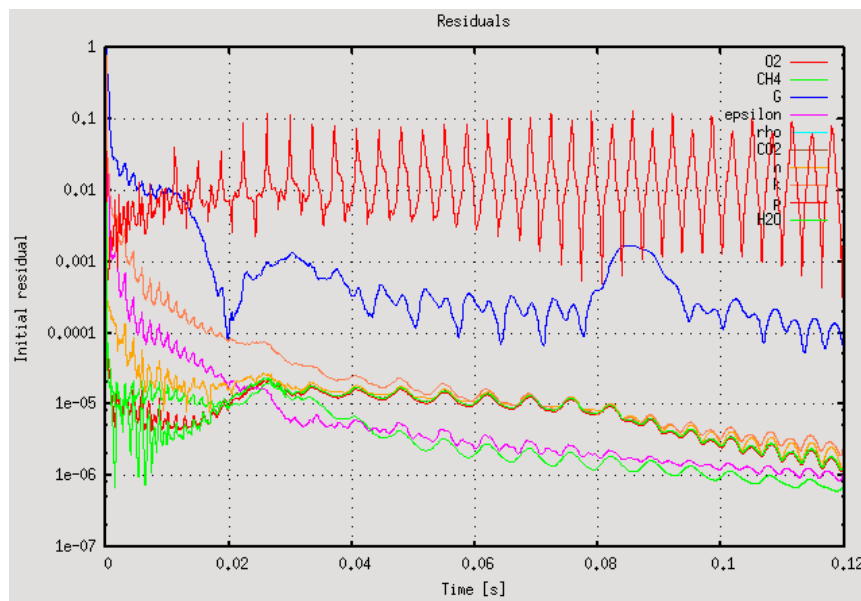


Figure 7.12: Residuals plot of the variables solved for during the Sandia Flame D simulation - GLB-C1.

In the reacting flow simulations the stopping criteria are set to be the Temperatures (max, min, volume average and area weighted average at the outlet). The temperature field converges after 0.12 seconds of physical time as seen in fig. 7.13 which corresponds to the computational time of 0.62 hours for mesh C1 using global reaction mechanism, PaSR combustion model and P1 radiation model. All the subsequent simulations for different grids and reaction mechanisms are thus run for 0.12 seconds of physical time and the computational times are recorded in table 7.3.

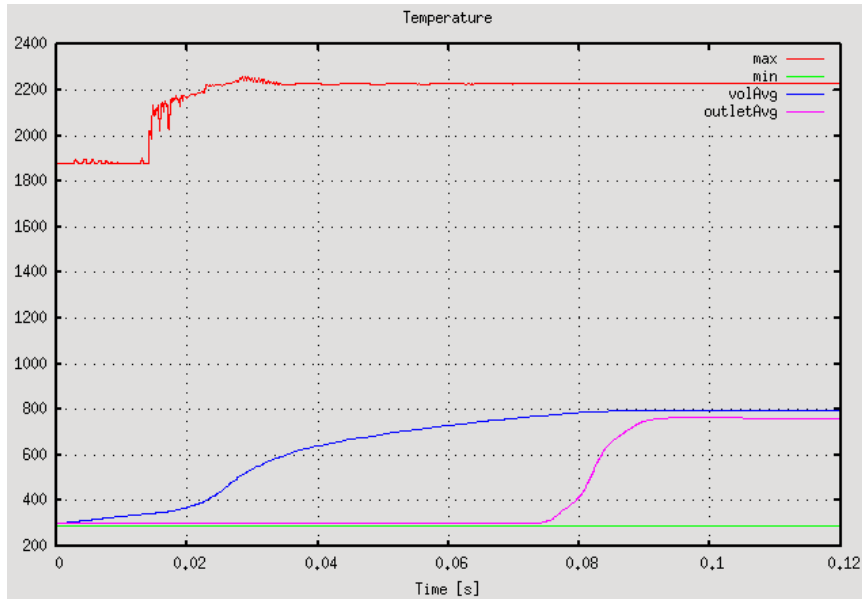


Figure 7.13: Plot of temperatures for the Sandia Flame D simulation - GLB-C1.

Test Case	Mesh C1 (# 5835)	Mesh R1 (# 23340)	Mesh R2 (# 45822)
GLB	0.62	3.58	16.73
2-Step	0.47	3.31	-
4-Step	1.67	7.77	-
GRI-Mech3.0	59.94	-	-

Table 7.3: Runtime(hours) on 4 cores (MPISSOR table C.1) using openMPI.

7.9 Summary

In general, there is a good qualitative agreement seen between the results from simulations and experimental data for the Sandia National Laboratories Flame D. Results for temperature and global species mass fractions are in good agreement with the experimental data. The observed discrepancies between the predictions and the experimental data for Sandia Flame D problem is attributed to the round-jet anomaly of the $k - \epsilon$ turbulence model as reported in literature [3] and the use of simplified P1 approximation model for radiative heat transfer calculations. Overall, the comparison gives a good indication of the adequacy and accuracy of the implemented solver furnaceFoam and its readiness for further combustion application development.

Chapter 8

Burner Flow Reactor

8.1 Introduction

The Burner Flow Reactor [5] is a co-fired coal biomass burner located at Brigham Young University (BYU) in Utah USA. It is an axi-symmetric long cylindrical vertical-fired reactor fig. 8.1 with a swirling flow. The Burner Flow Reactor combines many physical problems that need to be modeled of which the most challenging problem is to model coal combustion. Coal combustion is a complex process as it involves phase change due to devolatilization of coal particles. Further, the combustion of solids introduce the challenge of tracking particles in the flow domain. For modelling coal combustion, the first step is to develop a gas combustion model which is also the subject of the present chapter and subsequent chapters. Modelling gas combustion is relatively easier since it does not involve phase change. To model gas combustion in the Burner Flow Reactor geometry the new solver `furnaceFoam` is used in chapter 10 and the results are compared with the commercial package ANSYS Fluent. The turbulent reacting flow in the Burner Flow Reactor geometry has been analyzed in a previous work [8].

This chapter presents the geometry of the Burner Flow Reactor test case along with the boundary and initial conditions for isothermal and reacting flow simulations in the subsequent chapters. The sensitivity of OpenFOAM to mesh non-orthogonality is demonstrated in section 8.4 and the ways to correct the numerical errors due to mesh non-orthogonality are discussed. Finally the conclusions are presented in section 8.5.

8.2 Burner Flow Reactor Geometry

The dimensions of the furnace geometry are depicted in a 2D sketch in fig. 8.1. The location of the center, middle and secondary inlets is depicted in fig. 8.2. The primary fuel stream is injected through the center inlet, secondary fuel stream through the middle inlet and ambient air is injected through the secondary inlet.

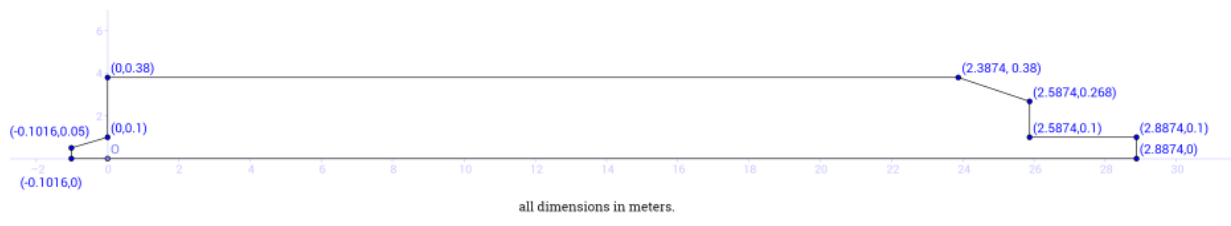


Figure 8.1: Sketch of internal dimensions of the Burner Flow Reactor

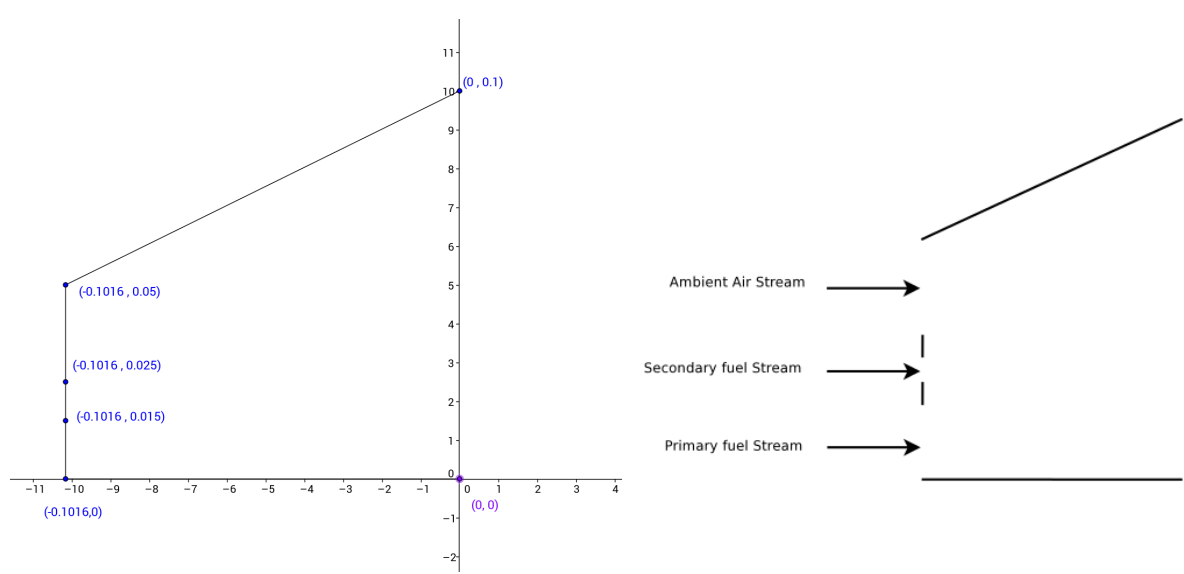


Figure 8.2: Center, middle and secondary inlet dimensions in meters.

8.3 Boundary and Initial Conditions

Dirichlet boundary condition is prescribed at the inlet for all variables except for pressure, that uses a Neumann boundary condition where the normal derivative i.e. gradient is zero. No-slip walls are used, together with a zero gradient boundary condition for the chemical species. The thermal boundary condition for the walls is adiabatic with constant temperature. The standard wall functions for turbulence kinetic energy and dissipation are used for near wall treatment. Pressure outlet boundary condition is used for the outlet where ambient conditions are prescribed. The detailed boundary conditions for the case are listed in table 8.1.

Domain	Pressure	Velocity	k	ϵ	T(K)	CH ₄	O ₂	N ₂
Primary Fuel	zG	(2.053, 0, 0)	0.0375	0.15	300	1.0	0	0
Secondary Fuel	zG	(4.827, 0, 0)	0.0375	0.15	300	1.0	0	0
Ambient Air	zG	(10, 4.5, 12)	0.0375	0.15	411	0	0.234	0.766
Furnace Walls	zG	(0, 0, 0)	zG	zG	550	zG	zG	zG
Outlet	1e+5	zG	zG	zG	zG	zG	zG	zG
Internal Volume Field Initialization	1e+5	cold flow profile	0.0375	0.15	1000	0	0.234	0.766

zG - zeroGradient.

Velocity expressed in (axial, radial, tangential) components.

k - Turbulent kinetic energy, ϵ - Turbulent dissipation

Mass fractions of species listed.

All quantities in SI units.

Table 8.1: Boundary and initial conditions for reacting flow simulations.

8.4 Computational Domain and Mesh Non-orthogonality Tests

The geometry of the Burner Flow Reactor is axi-symmetric and therefore, the problem is solved assuming symmetry about the center line. The computational domain is designed as a sector of 5° with imposed periodic boundary conditions to reduce the computational cost. The block structured mesh is made in Salome, an open source integration platform for numerical simulation. The mesh generated using Salome is converted from Ideas .unv format to OpenFOAM format using the utility ideasUnvToFoam provided by OpenFOAM. The mesh after being imported into OpenFOAM is rotated about the center line to generate a sector of 5° . The axi-symmetric mesh created as a wedge straddling along the X-Y plane with axis of the furnace along the X-axis and inlet of the furnace aligned with the Y-axis is shown in fig. 8.3.

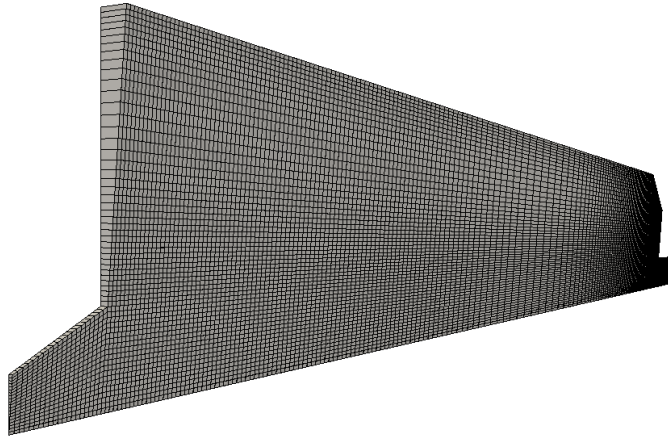


Figure 8.3: Axisymmetric 1-cell thick mesh with wedge angle(angle of cone) = 5° .

The geometry of the BFR is more complicated than the Sandia Flame D geometry fig. 7.2. It cannot be sub-divided into rectangular blocks and therefore orthogonality of the mesh cannot be ensured. Non-orthogonality is a significant contributor to error in OpenFOAM because of the way that gradients are calculated at cell faces. By default, gradients at cell faces are calculated by taking the difference between values at neighboring cell centers and dividing by the appropriate distance [55]. The non-collinearity between the center-to-center vector of adjacent cells and the normal vector of the face that connects these cells introduces significant errors during the computation of diffusive terms occurring in any partial differential equation [43].

The default approach used for approximating the diffusive flux D_f at the face f eq. (6.3) results in an approximation for the gradient in the direction of the vector connecting the two cell centers. This approach is not very good when the non-orthogonality of the grid is high. A common procedure used to correct for the mis-alignment between the line joining the cell centers and the face normal is to use non-orthogonality corrections. The orthogonal correction approach discussed earlier eq. (6.5) is one of the many approaches available in literature. The reader is referred to [42] [43] for more information on the common approaches used for non-orthogonality corrections.

It is possible to correct for non-orthogonality effects in OpenFOAM by making an appropriate choice of the numerical scheme used for calculating gradient at any cell face. The discretisation schemes used for calculating the surface normal gradient (snGrad) and laplacian terms are specified by the user in the fvSchemes file. A surface normal gradient (snGrad) evaluated at a cell face, is the component normal to the face, of the gradient of values at the centres of the 2 cells that the face connects. A surface normal gradient is also required to evaluate a laplacian term using Gaussian integration. The recommended discretisation scheme for snGrad based on the average non-orthogonality of the mesh [39] is listed in table 8.2.

Average Non-orthogonality	snGrad Scheme
Greater than 80	Redo Mesh
Between 70 and 80	limited 0.33
Between 50 and 70	limited 0.5
Less than 50	corrected
0	orthogonal

Table 8.2: Choice of snGrad scheme for different levels of non-orthogonality.

The OpenFOAM checkMesh utility when run on any mesh reports the level of average non-orthogonality to the user. The surface normal gradient calculation, snGrad, is integral to pressure-velocity algorithms on unstructured meshes and the choice of method has an important impact on both accuracy and stability.

Additional non-orthogonal corrections are required to correct for non-orthogonality effects in OpenFOAM. The number of non-orthogonal correctors are specified by the nNonOrthogonalCorrectors keyword in the fvSolution file. It specifies the number of times the pressure correction equation is solved in the pressure-velocity coupling algorithm used (SIMPLE/PISO/PIMPLE). The number of non-orthogonal correctors should correspond to the mesh for the case being solved. The following number of non-orthogonal corrections based on the average non-orthogonality of the mesh are recommended to the user [39] and are listed in table 8.3.

Average Non-orthogonality	nNonOrthogonalCorrectors
Greater than 80	Redo Mesh
Between 70 and 80	3
Between 50 and 70	2
Less than 50	1
0	0

Table 8.3: Choice of number of non-orthogonal corrections for different levels of non-orthogonality.

To illustrate the effect of mesh non-orthogonality on the numerical results we consider two different types of block-structured grids N1 and C1 with identical cell count. N1 is a non-orthogonal grid with significant non-orthogonality (44) near the inlet, whereas C1 is an almost orthogonal grid with zero non-orthogonality near the inlet as shown in fig. 8.6. The block decomposition of the grids C1 and N1 are shown in figures 8.4 and 8.5 respectively.



Figure 8.4: Grid C1 block decomposition



Figure 8.5: Grid N1 block decomposition

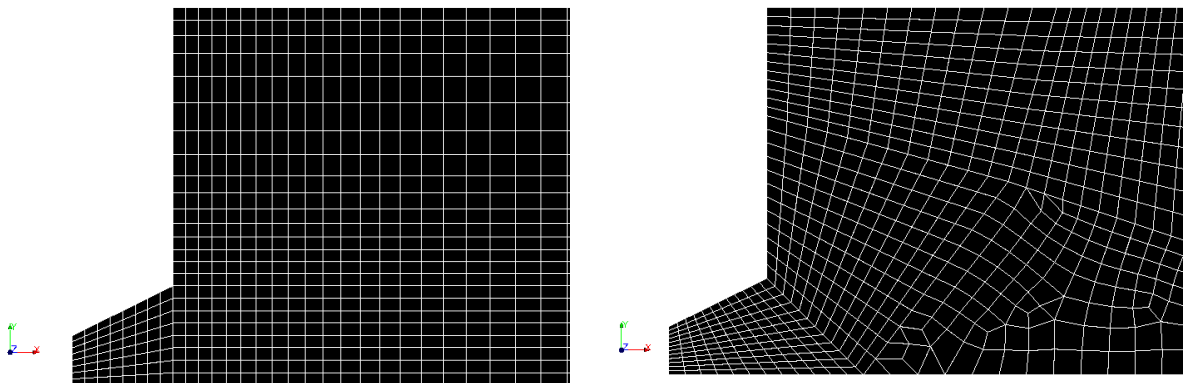


Figure 8.6: Mesh near the inlet for grids C1 and N1(right)

The incompressible isothermal flow simulations are done using the grids C1 and N1 and the flow near the inlet are compared in fig. 8.7. It can be observed that the flow is significantly deviated near the inlet for grid N1 due to the non-orthogonal cells near the corner. The errors due to non-orthogonality are minimized by using corrected schemes for the surface normal gradients and by doing additional non-orthogonal corrections in the SIMPLE algorithm used for pressure-velocity coupling. However in practice, there are significant errors when the angle between the face normal and cell-center vector is large and cannot be completely eliminated using non-orthogonal corrections as shown in fig. 8.8.

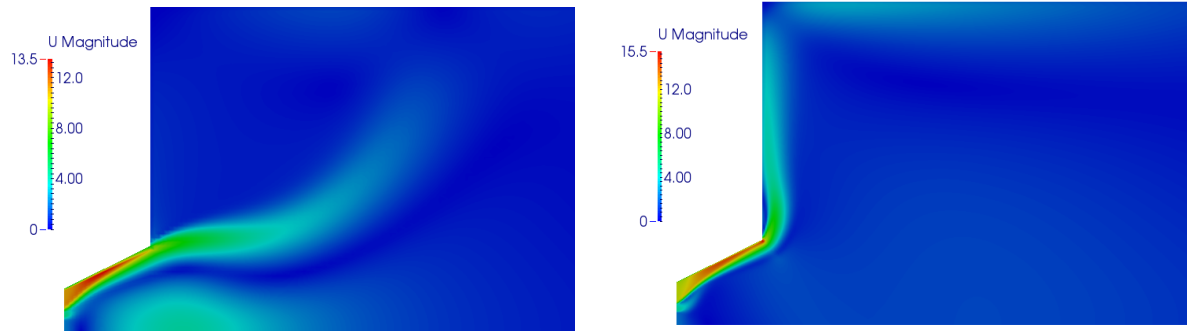


Figure 8.7: Velocity plot near the inlet for grids C1 and N1 (without orthogonal corrections)

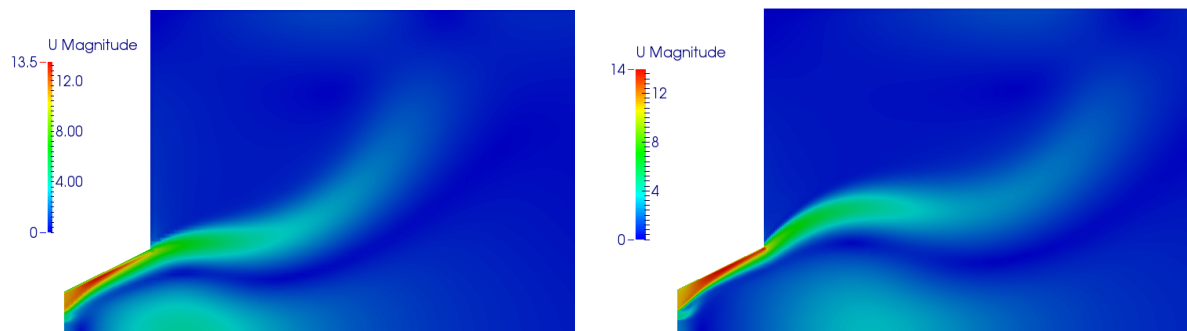


Figure 8.8: Velocity plot near the inlet for grids C1 and N1 (with orthogonal corrections)

The orthogonal grid C1 (cell count 11700) is further refined to obtain the grid R2 (cell count 46800) to be able to capture the steep gradients in chemical species concentrations for reacting flow simulations. All subsequent comparisons are done using grid R2 for both isothermal flow simulations in chapter 9 and reacting flow simulations in chapter 10.

8.5 Conclusions

From this analysis we conclude that OpenFOAM is sensitive to mesh non-orthogonality because of the way gradients are computed at the cell faces. Non-orthogonal corrections can significantly reduce the numerical errors in the calculation of diffusion terms but cannot completely eliminate them. Therefore, a quad-dominant mesh in 2D and hex-dominant mesh in 3D with minimum non-orthogonality is the preferred choice.

Chapter 9

Cold Flow Computations

9.1 Introduction

The purpose of the present chapter is to compare incompressible isothermal flow simulation results for BFR with results obtained using the commercial package ANSYS Fluent. The case setup is a comparison of the two codes using a steady-state incompressible turbulent flow solver. This is done to see how the turbulence models compare, if these are similar, later calculations using reactions and chemistry can rule the turbulence from any errors. The standard $k - \epsilon$ turbulence model 2.2.1 is considered for simplicity and the solver used in OpenFOAM is the steady-state incompressible turbulent flow solver simpleFoam 6.4. The detailed boundary conditions for the case are listed in table 8.1 with the exception that for isothermal flow computation the fluid considered is air with a constant kinematic viscosity of $1e-05 \text{ kg}/(\text{m}\cdot\text{s})$.

The simulations are done in ANSYS Fluent with the same initial and boundary conditions and turbulence model. To compare the isothermal flow simulation results the same mesh R2 generated in Salome is imported into both OpenFOAM and ANSYS Fluent. This is to ensure that any discrepancies in the simulation results are due to the different implementation of the turbulence models in OpenFOAM and ANSYS Fluent and are not due to different computational mesh.

9.2 Comparison of Results with ANSYS Fluent

The contour plot of the velocity magnitude obtained using the open-source, data analysis and visualization application ParaView is shown in fig. 9.1. From the figure a re-circulation zone close to the inlet can be observed where the velocity magnitudes are higher. The region close to the inlet is where the flow dynamics is active thus requiring finer mesh for resolving the turbulent eddies and is therefore also the computationally expensive zone.

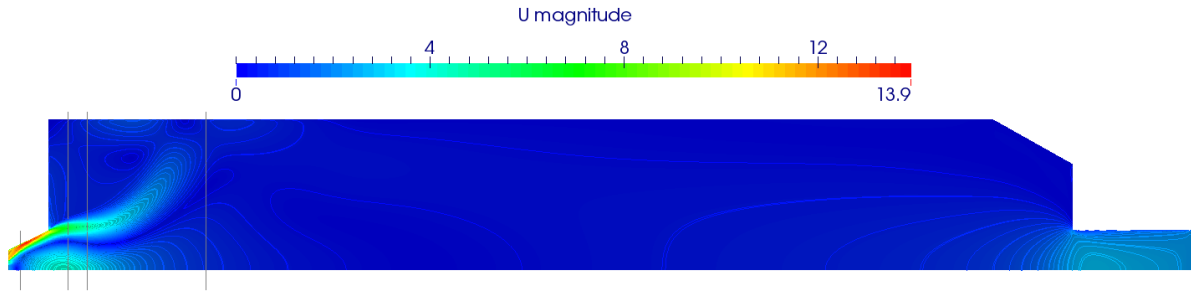


Figure 9.1: Contours of velocity magnitude(m/s) for cold flow simulation in OpenFOAM

To compare the two cases quantitatively, the data was exported from ANSYS Fluent and OpenFOAM into Matlab. The axial, radial and tangential velocity components across the furnace cross-section have been compared at varying distances from the inlet as shown in fig. 9.2, fig. 9.3 and fig. 9.4 respectively.

The comparison is made at axial distances of $x = 0.03\text{m}$, $x = 0.15\text{m}$, $x = 0.20\text{m}$ and $x = 0.50\text{m}$ from the inlet as shown by marked lines in fig. 9.1. From the plots it can be observed that the velocity's are higher close to the inlet and the flow downstream(0.5m from inlet) in the BFR has low velocities. The axial and radial components of velocity close to the inlet(0.15m and 0.2m) are negative which indicates the presence of recirculation zone close to the inlet. The tangential or swirl component of the velocity corresponding to the secondary air inlet is shown in fig. 9.4.

The active zone close to the inlet is where all the mixing takes place and is therefore the region where the reactions occur, illustrated later in chapter 10. The mixing process is further enhanced by the swirl component of the velocity.

9.3 Summary

The chapter has compared cold flow simulation results from OpenFOAM with ANSYS Fluent using the standard $k - \epsilon$ turbulence model for the BFR geometry as the test case. The isothermal flow results from OpenFOAM and ANSYS Fluent are seen to be in very good agreement. Only slight differences exists between the results at axial distance of 0.15 meters from the inlet where the flow dynamics is most active. Overall the two codes are considered to be on par with regards to the results, setup time and computation time (table 10.2).

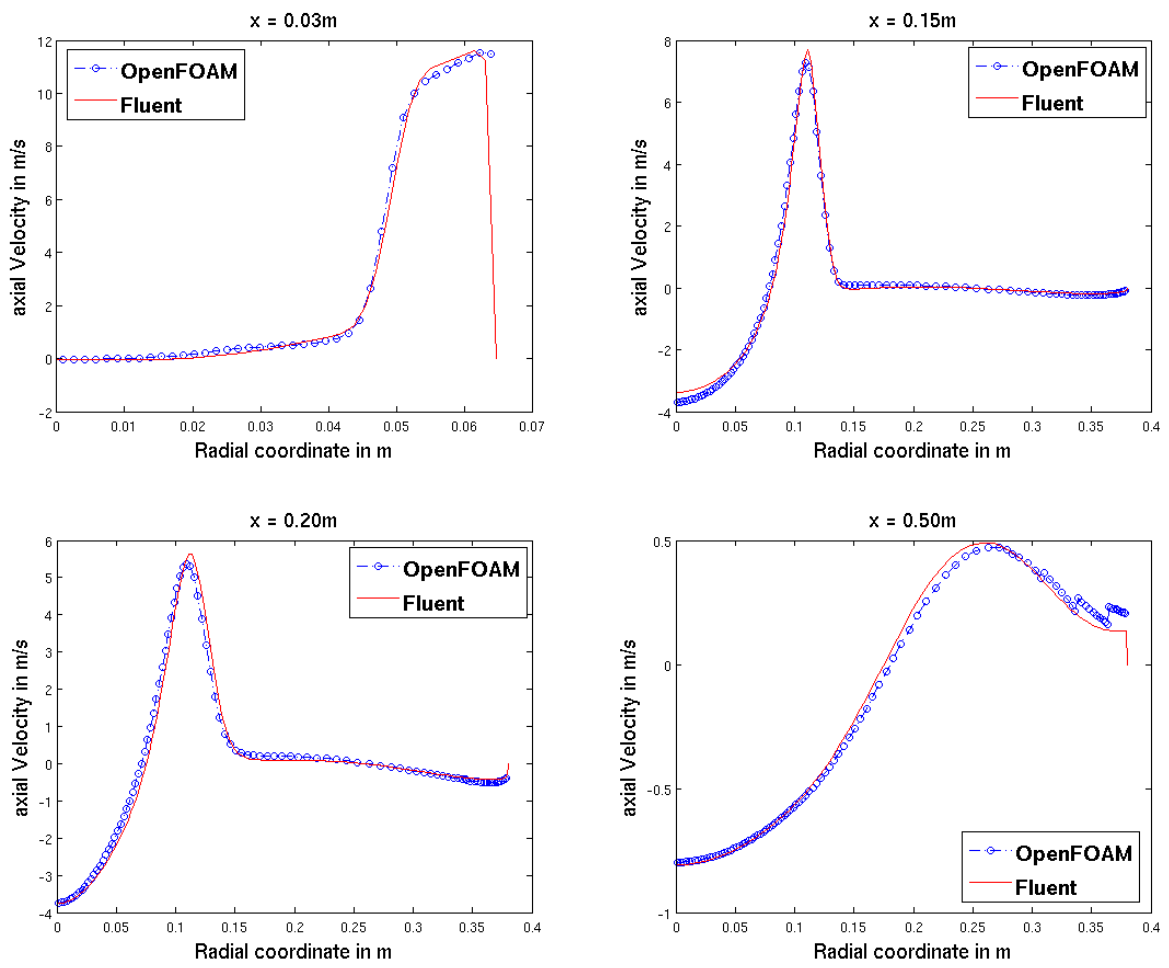


Figure 9.2: Axial velocity profile comparison of OpenFOAM and ANSYS Fluent

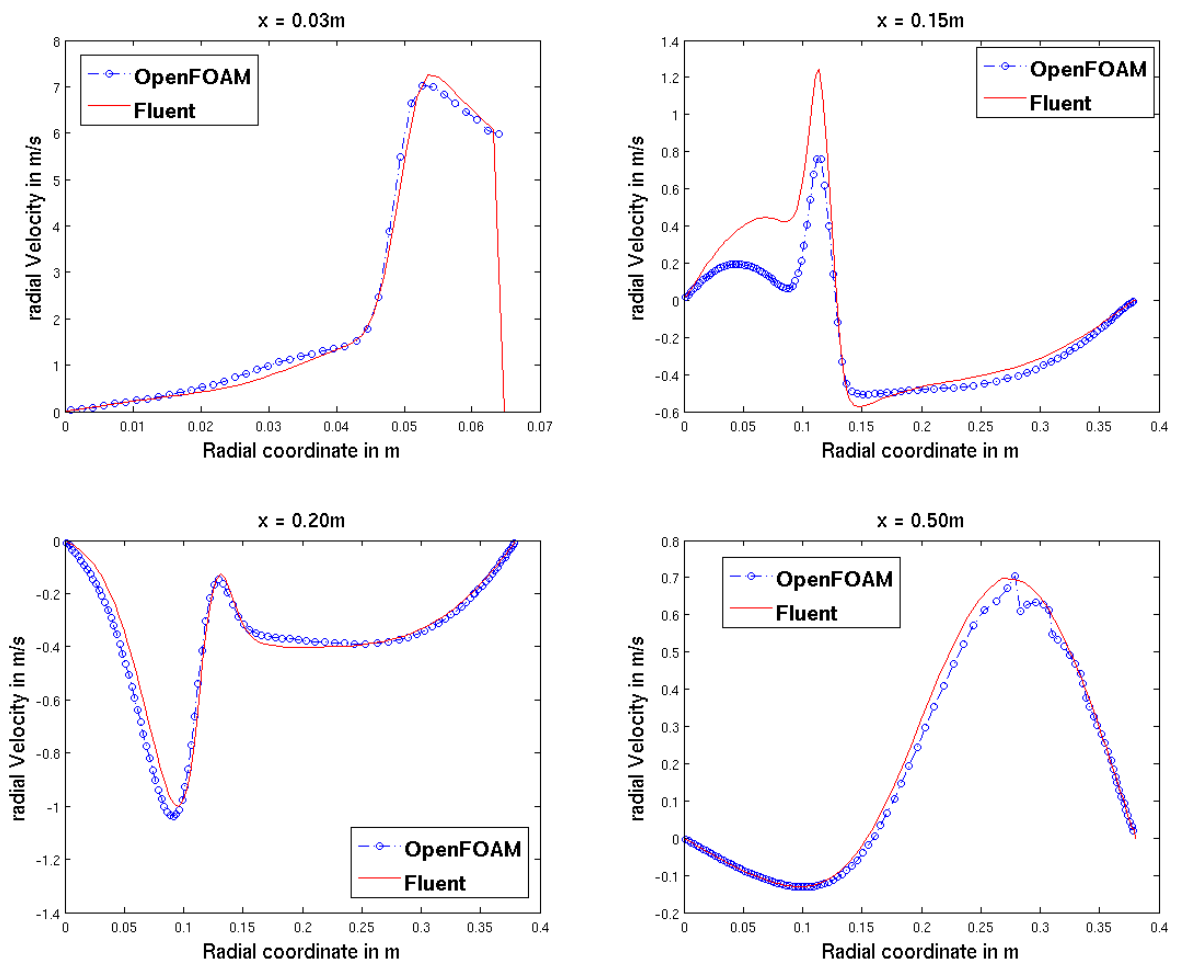


Figure 9.3: Radial velocity profile comparison of OpenFOAM and ANSYS Fluent

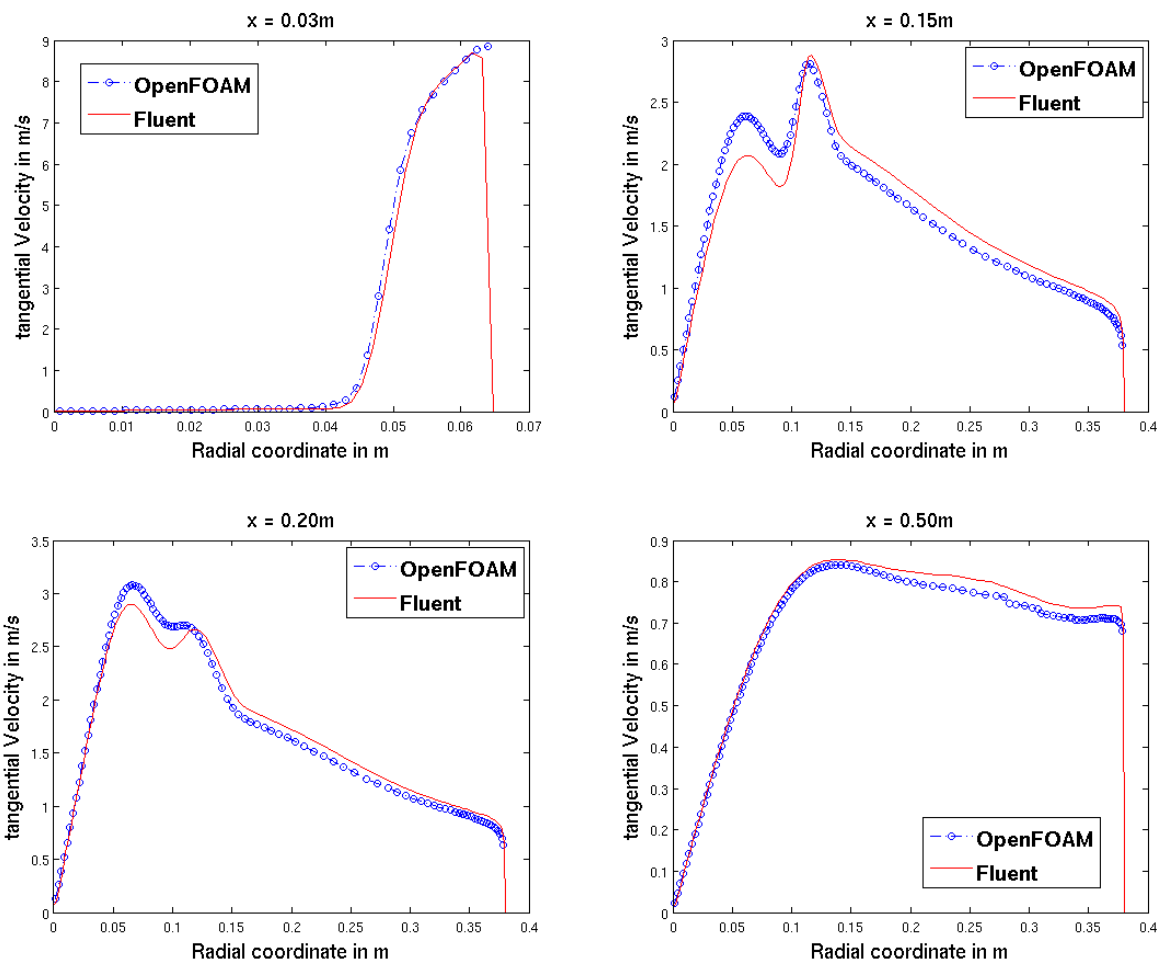


Figure 9.4: Tangential velocity profile comparison of OpenFOAM and ANSYS Fluent

Chapter 10

Reacting Flow Computations

10.1 Introduction

This chapter is an extension of the previous chapter to include combustion into the turbulent flow computations. The furnaceFoam solver (section 6.5) based on the partially stirred reactor (PaSR) combustion model is compared with the steady state solver for turbulent combustion in ANSYS Fluent. Since ANSYS Fluent has no implementation of the PaSR combustion model, the steady state solver based on the Eddy Dissipation Concept is used for comparison. The gaseous combustion of Methane with global 1-step reaction mechanism is considered for simplicity. The rate expression used for the global reaction is the same as for the Sandia Flame D test case 7.6. The detailed boundary conditions used for the reacting flow computations are listed in table 8.1.

The simulations are done in ANSYS Fluent with the same initial and boundary conditions and turbulence model. To compare the reacting flow simulation results the same mesh R2 generated in Salome is imported into both OpenFOAM and ANSYS Fluent.

10.2 Equivalence Ratio Calculations

In combustion calculations we often perform a simple analysis which assumes that the fuel burns completely to form products (complete combustion). We can also calculate the air to fuel ratio necessary for complete combustion, which is called the stoichiometric air/fuel ratio. Combustion with less air than the stoichiometric air requirement is called fuel rich, and combustion with air in excess of the stoichiometric air requirement is called lean combustion. The equivalence ratio is used in combustion calculations to define the strength of a mixture with respect to the stoichiometric mixture strength. The equivalence ratio ϕ is defined as

$$\phi = \frac{(A/F)_{st}}{(A/F)_{ac}}$$

If 1 mole fuel requires x moles O_2 for complete combustion and assuming air (molar mass 28.97g/mol) is 79% N_2 and 21% O_2 we have

$$(A/F)_{st} = \frac{\frac{79}{21}M_{N_2} + M_{O_2}}{\frac{1}{x}M_{fuel}} \quad (10.1)$$

Combustion of CH_4 with the global reaction mechanism $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$ yields $(A/F)_{st} = 17.13$. In the actual mixture the global air/fuel ratio can be calculated as the ratio of total mass flow rate of air to fuel. For the BFR using the ideal gas law we can obtain the total mass flow rate of air as $\sum_{\#inlets} \frac{PM_{air}}{RT} \vec{A} \cdot \vec{U}$ and fuel as $\sum_{\#inlets} \frac{PM_{fuel}}{RT} \vec{A} \cdot \vec{U}$.

Since the Burner Flow Reactor geometry is axi-symmetric and is considered as a wedge, it must be taken into account during equivalence ratio calculations. Indexing the primary fuel, secondary fuel and ambient air inlets as 1, 2 and 3 we have

$$\sum_{\#inlets} \frac{PM_{fuel}}{RT} \vec{A} \cdot \vec{U} = \frac{PM_{CH_4}}{RT_1} \pi R_1^2 U_1 + \frac{PM_{CH_4}}{RT_2} \pi (R_2^2 - R_1^2) U_2$$

$$\sum_{\#inlets} \frac{PM_{air}}{RT} \vec{A} \cdot \vec{U} = \frac{PM_{air}}{RT_3} \pi (R_3^2 - R_2^2) U_3$$

where U_1 , U_2 and U_3 are components of the velocity normal to the inlet patch given in table 8.1 and R_1 , R_2 and R_3 are the radius of primary fuel inlet, secondary fuel inlet and ambient air inlet respectively as shown in fig. 8.2. Therefore we have

$$(A/F)_{ac} = \frac{\frac{PM_{air}}{RT_3} \pi (R_3^2 - R_2^2) U_3}{\frac{PM_{CH_4}}{RT_1} \pi R_1^2 U_1 + \frac{PM_{CH_4}}{RT_2} \pi (R_2^2 - R_1^2) U_2} \quad (10.2)$$

and global equivalence ratio calculated for the inflow boundary conditions listed in table 8.1 is $\phi = 1.65$. A value greater than 1 indicates that the mixture is fuel rich.

10.3 Comparison of Results with ANSYS Fluent

The contour plot of the velocity magnitude obtained using ParaView is shown in fig. 10.1. From the figure a re-circulation zone close to the inlet similar to that of the cold flow in fig. 9.1 can be observed where the velocity magnitudes are higher. The active zone close to the inlet is where all the mixing takes place as shown in fig. 10.2 and is therefore the region where the reactions occur.

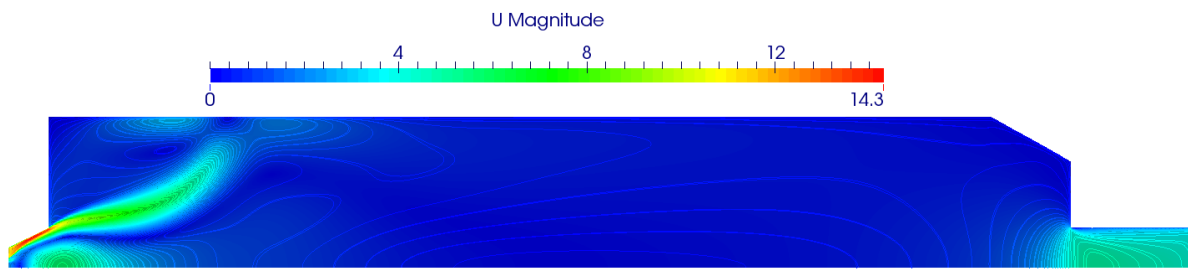


Figure 10.1: Contours of velocity magnitude (m/s)

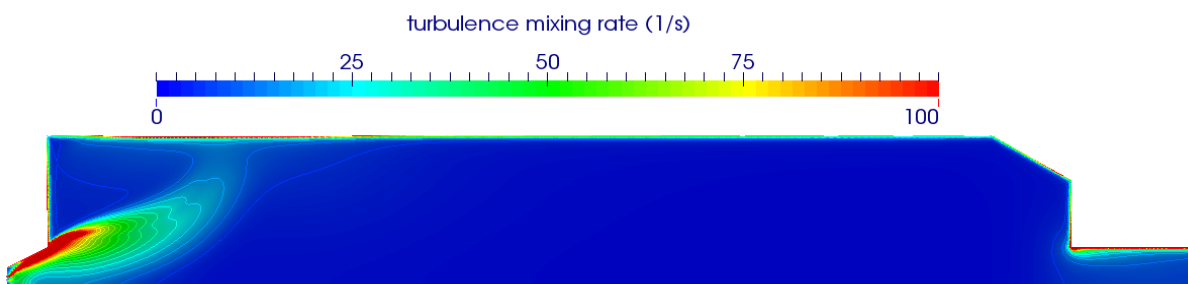


Figure 10.2: Contours of turbulence mixing rate ($\frac{\epsilon}{k}$)

The contour plot of the temperature is shown in fig. 10.3. From the plot it can be observed that the temperatures close to the inlet are very high, since turbulent combustion is mixing dominated and the reactions occur in the region close to the inlet where mixing is high.

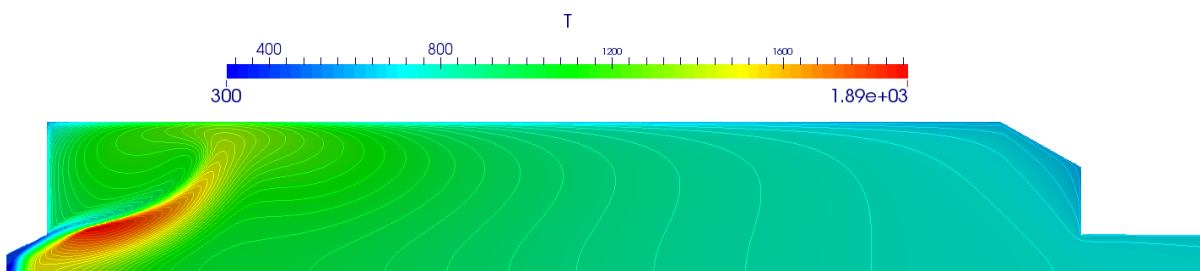


Figure 10.3: Contours of temperature (K)

From the contour plots for velocity and temperature it can be concluded that.

- Reactions occur near the inlet where mixing is high
- Flow downstream in the BFR has low velocities and no reactions.

To compare the two cases quantitatively, the data was exported from ANSYS Fluent and OpenFOAM into Matlab. Temperatures are plotted along the axis of the furnace as shown in fig. 10.4. The temperature profile far from the inlet is in very good agreement with the results from ANSYS Fluent. However temperature profile close to the inlet at an axial distance of 0.1-0.3m showed significant deviations. The discrepancies in the results are attributed to the fact that the reacting zone in the furnace is located close to the inlet where the choice of the turbulence-chemistry interaction model plays an important role. The temperatures predicted by the EDC model in ANSYS Fluent are found to be significantly higher than predicted by the PaSR model in OpenFOAM close to the inlet zone.

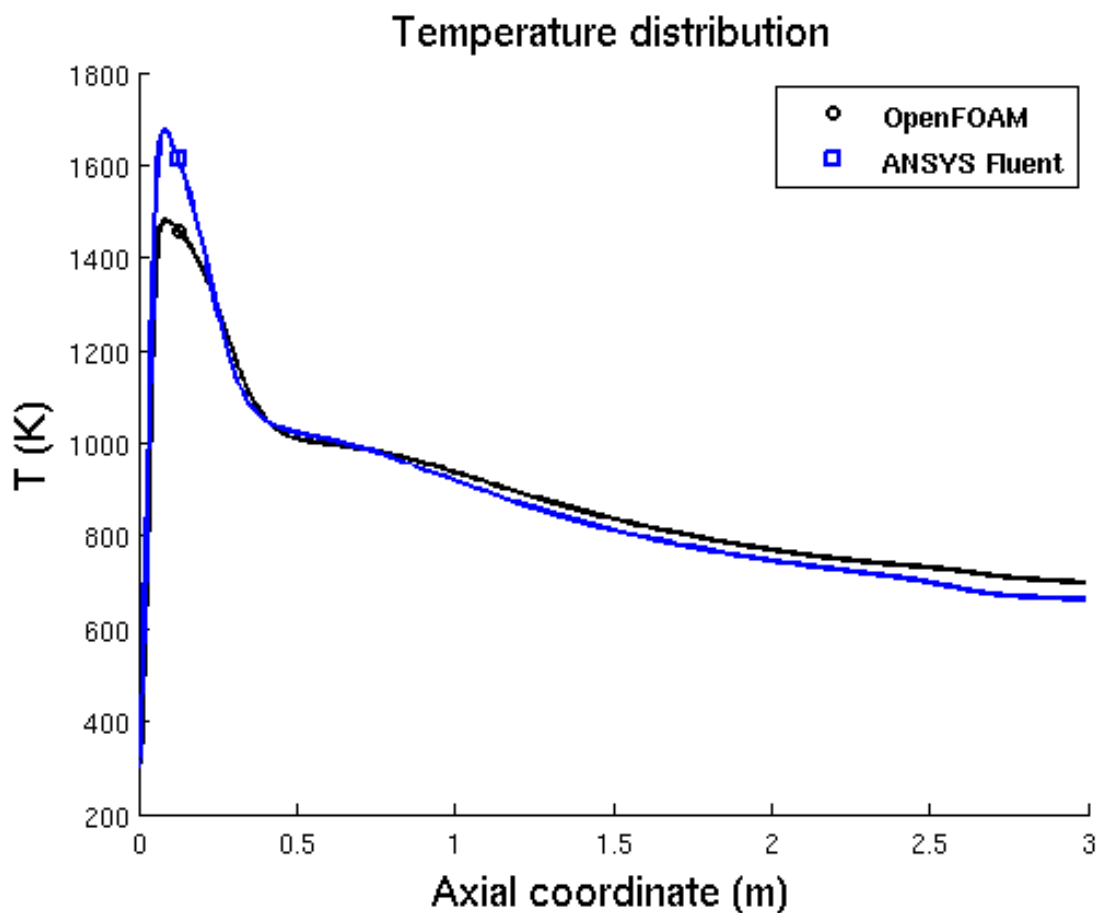


Figure 10.4: Temperature(K) profile along the central axis of the furnace.

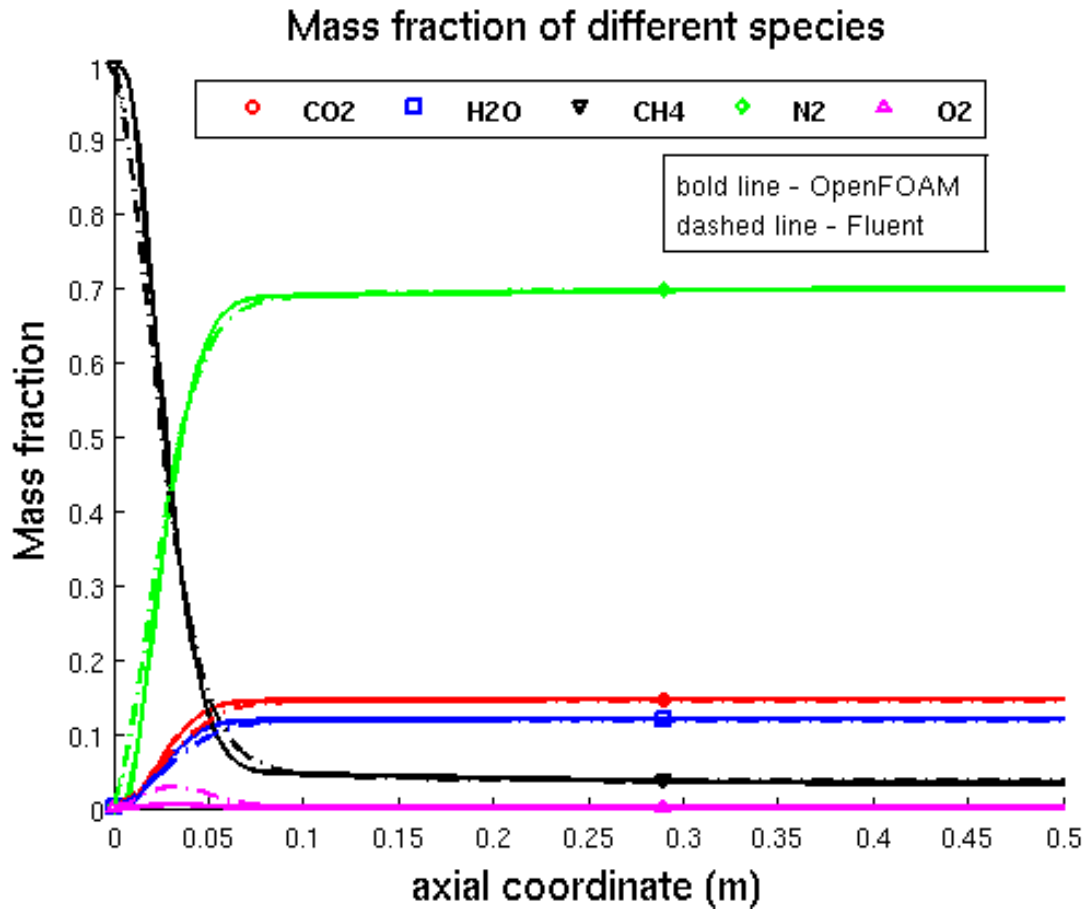


Figure 10.5: Mass fractions along the central axis of the furnace ($0 \leq x \leq 0.5$).

The plot of the mass fractions for different species along the axis of the furnace showed good agreement with ANSYS Fluent fig. 10.5. From the plot it can be observed that the mass fraction of the fuel CH_4 decreases along the central axis of the furnace whereas the mass fractions of the products CO_2 and H_2O increases. The mass fraction of O_2 remains almost zero on the axis because the air enters the domain through the secondary air inlet and also the mixture is fuel rich and therefore, all the oxygen in the reaction chamber is consumed. It is also interesting to see that the mass fractions change rapidly close to the inlet ($\leq 0.2m$) where the mixing is high and thus most of the reactions takes place close to the inlet. The area weighted mass fractions of the species at the outlet for both the cases are also presented in table 10.1.

Specie	OpenFOAM(PaSR)	ANSYS Fluent(EDC)
CO2	0.1467	0.1455
H2O	0.1201	0.1191
CH4	0.0339	0.0353
N2	0.6990	0.6982
O2	0.0001	0.0016

Table 10.1: Mass fractions of species at the outlet

10.4 Comparison of Infinitely fast chemistry model and PaSR model

In this section the results of the infinite fast chemistry model are compared to the PaSR detailed chemistry model. The infinitelyFastChemistry model in OpenFOAM is based on the simple assumption of “mixed is burnt” i.e. the chemical species react completely and form products as soon as they are mixed. The model does not take into consideration the chemical kinetics which means that the chemical time scale is neglected with respect to the flow time scale. The infinitelyFastChemistry model implementation in OpenFOAM does not take into account the turbulence mixing time in calculating the mean reaction rate (section 6.5). It is therefore different from the popular eddy break-up equilibrium chemistry model which is also based on “mixed is burnt” assumption but determines the reaction rate from the turbulence mixing time (section 2.5.1).

10.4.1 2D Axi-symmetric Simulations

The infinitelyFastChemistry model is used for simulating the Burner Flow Reactor in 2D and the predicted temperature distribution is compared to the PaSR model in fig. 10.6.

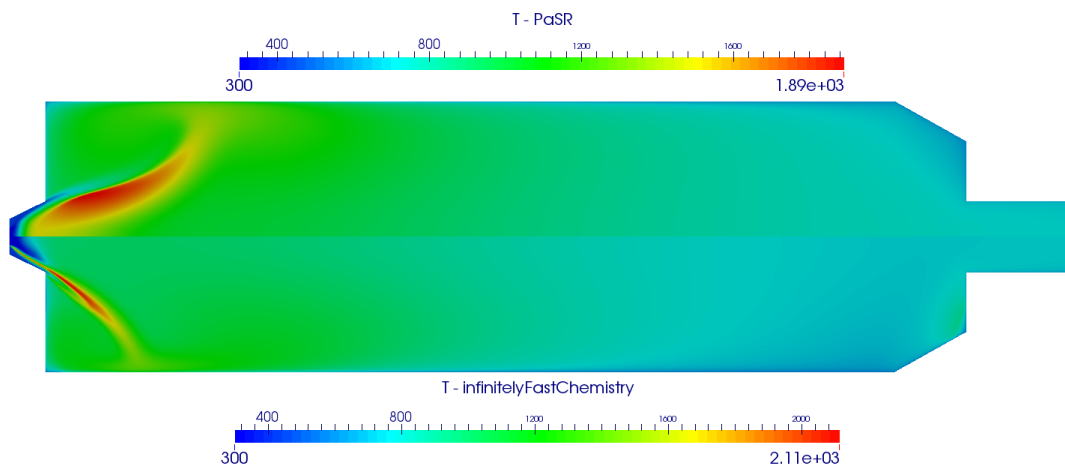


Figure 10.6: Contours of temperature (K)

The following two observations can be made from fig. 10.6.

- The temperatures predicted by the infinitelyFastChemistry model are considerably higher than predicted by the PaSR model. This was expected because the infinite fast chemistry model assumes that the chemical species react as soon as they are mixed and the reactions go to completion. Whereas in the PaSR model the reactions are kinetically inhibited by the Arrhenius law and do not go to completion thereby releasing less heat and therefore lower temperatures.
- It can also be observed from the temperature plots for infinitelyFastChemistry model that the flame is ignited at the inlet as soon as the fuel and oxygen enter the combustion chamber. This limits the use of the infinitelyFastChemistry model for combustion problems such as the Sandia Flame D where the fuel is partially premixed and the ignition takes place in the pre-inlet pipe itself before the fuel enters the combustion chamber. A solution is to compute both the Arrhenius rate and the mixing rate and use the smaller of the two in determining the reaction rate. This is implemented in the Finite-Rate/Eddy-Dissipation model of ANSYS Fluent.

10.4.2 3D Simulations

The mesh C1 is rotated by 360° in steps of 10° to obtain the mesh C13D with control volume count 421200. The 3D simulations are carried out using the infinitely fast approach with global chemistry. The radiation model used is finite volume discrete ordinates method (fvDOM). The primary ParaView filters Glyph and StreamTracer for creating Streamlines are used to obtain the flow field visualization in fig. 10.7.

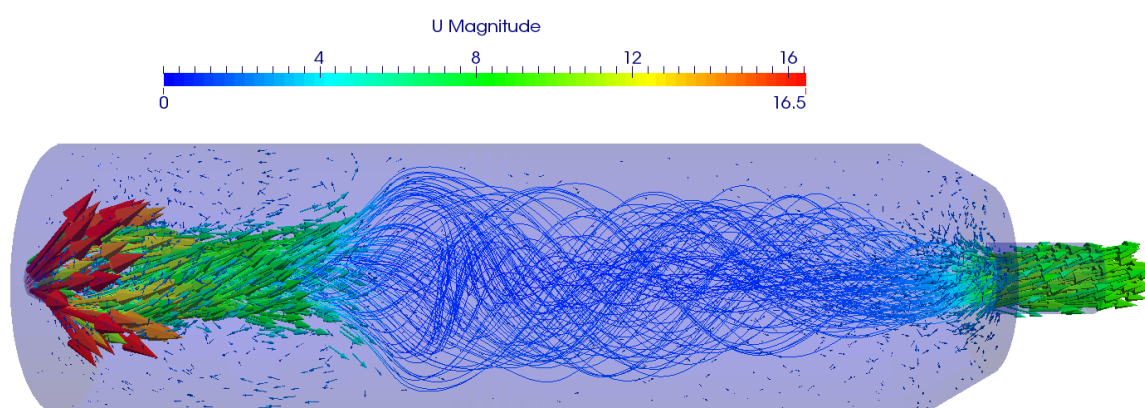


Figure 10.7: Glyph and StreamTracer filters applied on the velocity vector field

10.5 Computational Time

The temperature field converges after 4 seconds of physical time as observed in fig. 10.9 which corresponds to the computational time of 32.68 hours for mesh R2 using global reaction mechanism (GLB), PaSR combustion model and P1 radiation model. It can be observed from the temperature plots that the mixture ignites after 0.4 seconds when the maximum temperature increases from 1000K (initialized field) to 2500K. After 2 seconds of simulation the radiative heat transfer is enabled which causes a sudden drop in the volume average temperature as seen in fig. 10.9. All the subsequent simulations for different grids and reaction mechanisms are run for 4 seconds of physical time and the computational times are recorded in table 10.2.

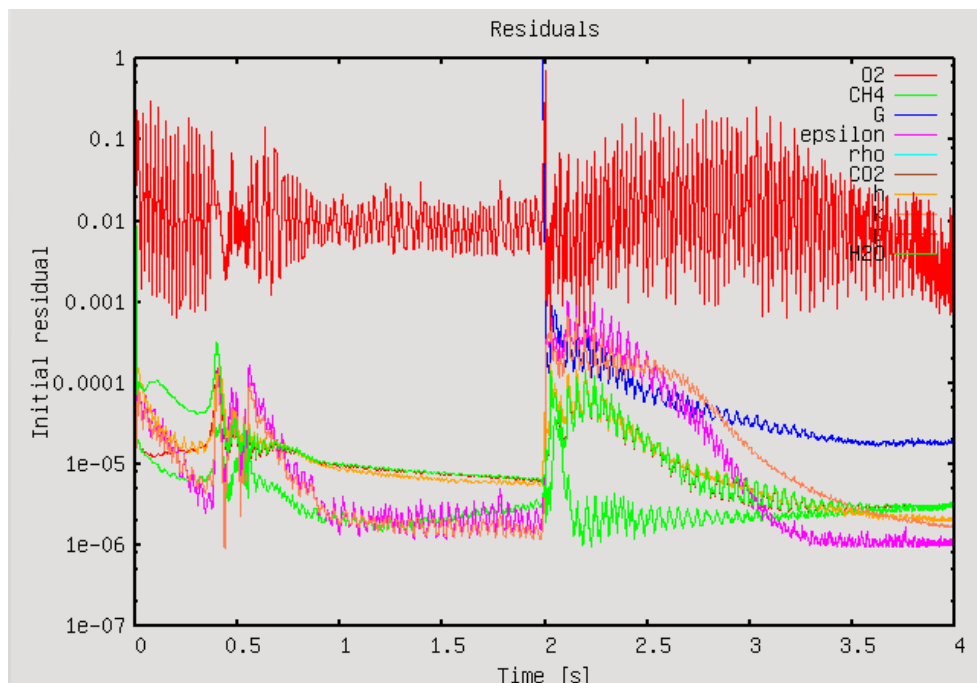


Figure 10.8: Residual plot of the variables solved for during the Burner Flow Reactor simulation $OF - R2 - GLB - P1$.

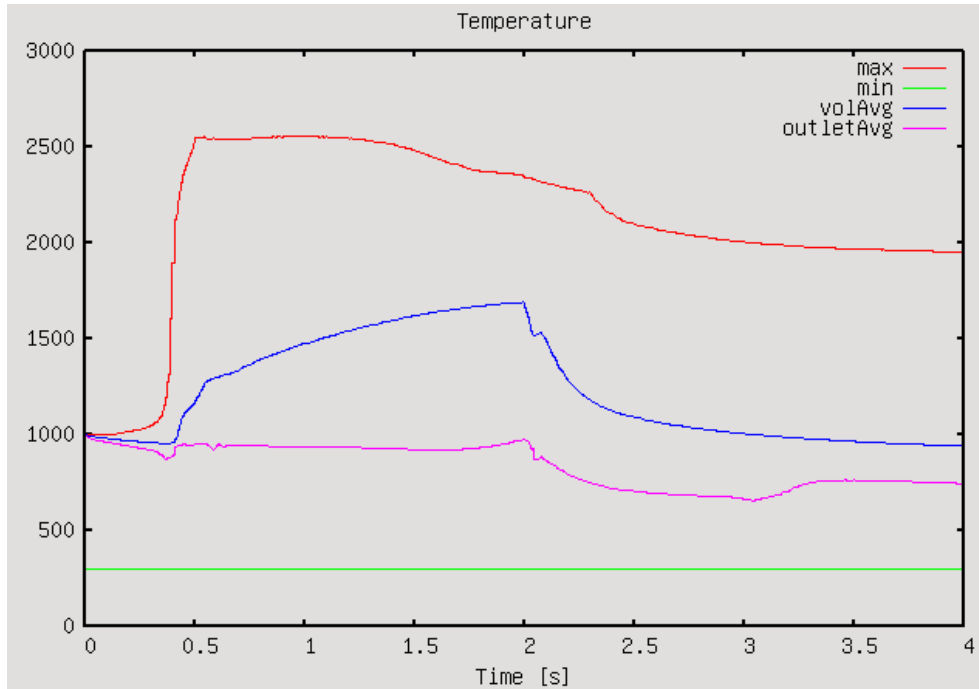


Figure 10.9: Plot of the temperatures for the Burner Flow Reactor simulation *OF – R2 – GLB – P1*.

Domain Decomposition

The simulations are run on the node MPISSOR (table C.1) with 4 cores. The OpenFOAM `decomposePar` utility is used for decomposing the computational domain into four sub-domains. The `decomposePar` utility reads the `decomposeParDict` dictionary file provided by the user to do the decomposition. The geometry and fields are broken up according to a set of parameters specified in the `decomposeParDict` dictionary file. The necessary parameters required are 1. The method of decomposition and 2. The number of sub-domains required. In the present work the type of decomposition used was “simple”, in which the domain is split into sub-domains by direction. The number of sub-domains in the -x , -y and -z directions are provided as input to the “simple” geometric decomposition method. Since the flame zone is located close to the inlet and spans the area between the axis and the furnace wall as shown in fig. 10.3 the vertical decomposition (fig. 10.10) is preferred over horizontal decomposition for the 2D Axi symmetric simulations. This is to ensure that the computationally expensive flame zone is equally distributed among the processors. For BFR simulations in 3D the domain is decomposed into 2 sub-domains in the -y direction and 2 sub-domains in the -z direction with 1 sub-domain along the -x direction. The resulting decomposition is shown in fig. 10.11.

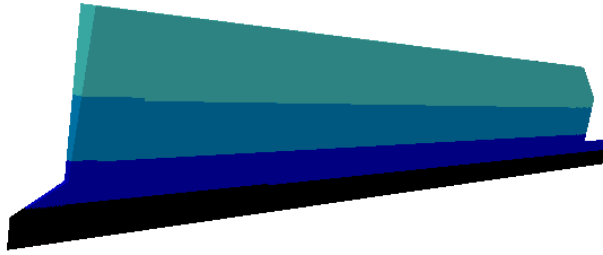


Figure 10.10: Domain decomposition for 2D Axisymmetric BFR simulations.

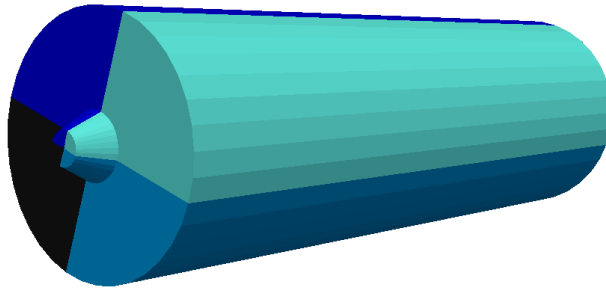


Figure 10.11: Domain decomposition for 3D BFR simulations.

Following conclusions can be drawn from the run time statistics presented in table 10.2.

- The steady state simulations in ANSYS Fluent ($AF - R2 - GLB - P1$) are considerably faster than the transient simulations in OpenFOAM. Besides a steady state solver in ANSYS Fluent a second reason for the speed-up is the In-Situ Adaptive Tabulation (ISAT) algorithm used in ANSYS Fluent for accelerating the detailed chemistry calculations. A speedup of two to three orders of magnitude can be attained by using the ISAT algorithm [13].
- The infinitelyFastChemistry simulations ($OF - R2 - Inf - P1$) are considerably faster than the detailed chemistry simulations ($OF - R2 - GLB - P1$) because of the computationally expensive Arrhenius chemistry calculations not being considered in the former.

Test Case	Isothermal Flow	Reacting Flow
<i>AF – R2 – GLB – P1</i>	~ 0.1	~ 3
<i>OF – R2 – GLB – P1</i>	0.12	32.68
<i>OF – R2 – Inf – P1</i>	0.12	22.55
<i>OF – C13D – Inf – fvDOM</i>	0.83	82.01

Table 10.2: Runtime(hours) on 4 cores (MPISSOR table C.1) using openMPI.

10.6 Summary

The chapter compares the reacting flow simulation results for the BFR geometry, obtained using `furnaceFoam` with the commercial package ANSYS Fluent. The implemented solver `furnaceFoam` is a transient solver based on the partially stirred reactor (PaSR) combustion model. Since PaSR combustion model is not available in ANSYS Fluent, the results were compared using the eddy dissipation concept model. The results obtained using OpenFOAM showed varying agreement with ANSYS Fluent. The mass fractions of species showed good agreement with ANSYS Fluent but the temperature profile showed some deviations close to the inlet. The differences in the simulation results are attributed to the different combustion models used in OpenFOAM and ANSYS Fluent. Overall, the comparison gives a good indication of the adequacy of the implemented solver `furnaceFoam` for tubulent combustion simulations. However the current bottleneck identified with OpenFOAM is the very high computational cost of the transient solver `furnaceFoam` as compared to the steady-state solver in ANSYS Fluent.

Chapter 11

NOx Reduction Mechanisms

11.1 Validation of NOxFoam using ANSYS Fluent

The solver NOxFoam for thermal NO prediction in industrial furnaces was described in detail in section 6.3. The following quantities from the main combustion calculations are required as input for NOx post-processing.

- Temperature(T) distribution.
- The velocity field (\vec{u}).
- The mass fraction Y_s of the participating chemical species and pressure(p) distribution.
- Turbulent viscosity μ_t .

The sensitivity of thermal NO formation rate to temperature distribution is evident from fig. 5.1. Therefore, identical temperature distribution have to be provided as input to the NOx post-processing utility's in ANSYS Fluent and OpenFOAM if they are to be compared. A possible solution to the problem is to import OpenFOAM data into ANSYS Fluent or vice-versa. There exist utility's foamMeshToFluent and foamDataToFluent for converting OpenFOAM mesh and data to ANSYS Fluent mesh and data formats respectively. The Sandia Flame D and the Burner Flow Reactor test problems were considered as 2D axi-symmetric problems and represented as wedge (fig. 7.3, fig. 8.3) in OpenFOAM. A 2D axi-symmetric wedge geometry in OpenFOAM when imported into ANSYS Fluent is considered as a 3D problem and therefore cannot be used for 2D axi-symmetric simulations.

Since importing OpenFOAM mesh and data into ANSYS Fluent is not a viable option for our case, an alternative solution is to solve the problems independently in both OpenFOAM and ANSYS Fluent and generate identical inputs for the NOx post-processors for further comparison. To be able to generate identical inputs it would be essential to exclude the complex phenomenon of combustion and radiation to have a similar temperature distribution. Since the high temperatures required for thermal NO formation are achieved after

combustion, the benchmark problems Sandia Flame D and Burner Flow Reactor cannot be directly used for validation.

To circumvent this problem the Burner Flow Reactor geometry was used with very high temperatures prescribed at the inlet (2200 K). The high temperatures and chemical species at the inlet are convected downstream into the furnace due to flow and turbulence. The reactions, combustion and radiative heat transfer phenomenon are disabled to exclude these complex effects from influencing the temperature distribution and thus the NOx chemistry. The detailed boundary conditions for the case are the same as for reactingFlow simulations (table 8.1) with the differences highlighted in table 11.1. The inputs to the NOx post-processors in OpenFOAM and ANSYS Fluent are compared in fig. 11.1

The area weighted average of the thermal NO mass fraction at the outlet is monitored and is taken as the convergence criterion. The thermal NO measurements for BFR geometry are made after 40 seconds of real time simulation when the NO mass fraction at the outlet has converged. The same criterion is used for all subsequent NO measurements in this chapter.

Domain	Pressure	Velocity	k	ϵ	T(K)	CH ₄	O ₂	N ₂
Primary Fuel	zG	(20, 0, 0)	0.0375	0.15	2200	1.0	0	0
Secondary Fuel	zG	(20, 0, 0)	0.0375	0.15	2200	1.0	0	0
Ambient Air	zG	(20, 0, 0)	0.0375	0.15	1000	0	0.234	0.766
Furnace Walls	zG	(0, 0, 0)	zG	zG	550	zG	zG	zG
Outlet	1e+5	zG	zG	zG	zG	zG	zG	zG
Internal Volume Field	1e+5	cold flow profile	0.0375	0.15	2000	0	0.234	0.766

zG - zeroGradient.

Velocity expressed in (axial, radial, tangential) components.

k - Turbulent kinetic energy, ϵ - Turbulent dissipation

Mass fractions of species listed.

All quantities in SI units.

Table 11.1: Boundary and initial conditions for NOxFoam validation test case.

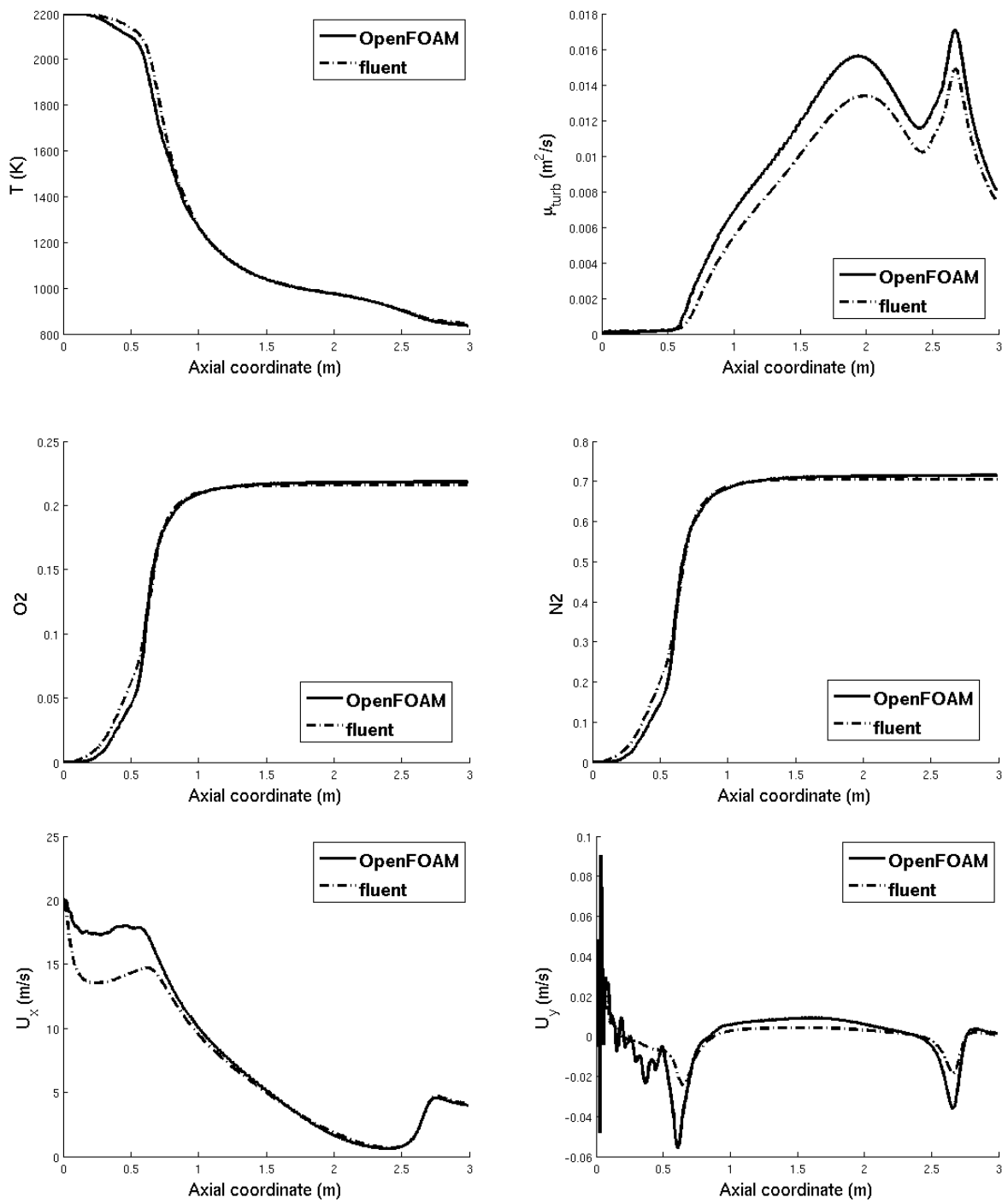


Figure 11.1: Inputs to NOx post-processors in OpenFOAM and ANSYS Fluent.

The thermal NO mass fraction along the central axis of the furnace are compared in fig. 11.2. The thermal NO measurement results are identical for OpenFOAM and ANSYS Fluent far from the inlet. Close to the inlet where the flow dynamics is active the agreement is good considering the fact that NO present in the furnace is in trace amounts when compared to the global species concentration. The discrepancies in the results can be attributed to the differences in the input flow profile as seen in fig. 11.1 for the axial component of the velocity field.

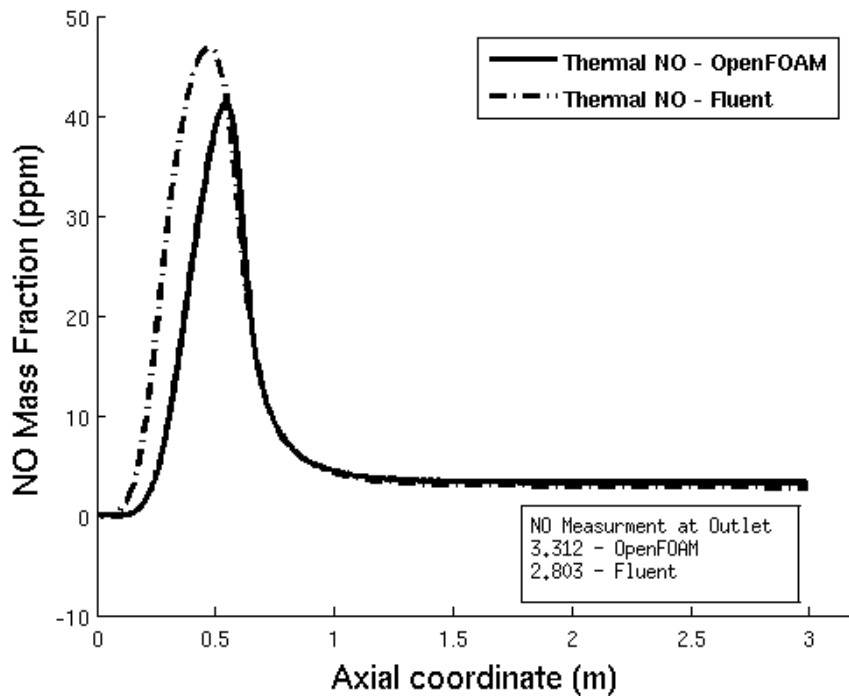


Figure 11.2: Thermal NO mass fraction (in ppm) along the central axis of the furnace.

11.2 Thermal NO Reduction

The primary anthropogenic source of NO_x is combustion. Since NO is the predominant NO_x specie formed during combustion [33], it is of utmost importance to predict NO formation in any combustion equipment and to devise methods for its reduction. In this section we state some of the most common methods used in industry for NO_x reduction. We illustrate the effectiveness of few of these methods using the Burner Flow Reactor geometry. In this context we shall restrict ourselves to only thermal NO reduction mechanisms.

In chapter 5 we illustrated the extended Zeldovich reaction mechanism eq. (5.1) governing thermal NO formation. Substituting the expressions for the partial equilibrium concentrations of O atoms eq. (5.7) in the rate expression governing the formation of NO eq. (5.5) we obtain

$$\frac{d[NO]}{dt} = 1.32 \times 10^{10} e^{-65493/T} T^{1/2} [O_2]^{1/2} [N_2] \frac{\left(1 - \frac{k_{b1} k_{b2} [NO]^2}{k_{f1} k_{f2} [N_2] [O_2]}\right)}{\left(1 + \frac{k_{b1} [NO]}{k_{f2} [O_2] + k_{f3} [OH]}\right)} \quad (11.1)$$

From the above condition it can be concluded that

$$\left.\frac{d[NO]}{dt}\right|_{max} = 1.32 \times 10^{10} e^{-65493/T} T^{1/2} [O_2]^{1/2} [N_2] \quad (mol/m^3 - s) \quad (11.2)$$

The strong dependence of the thermal NO formation on the combustion temperature and the lesser dependence on the oxygen concentration are evident from eq. (11.2). Thus, the best practical means of controlling NO is to reduce the combustion gas temperature and, to a lesser extent, the oxygen concentration [15].

When a fuel burns completely under constant pressure and if no external heat or work transfer takes place then all the energy liberated by the chemical reactions is used in heating the system. This achieves the highest possible temperature called the adiabatic flame temperature. The adiabatic flame temperature for methane combustion is calculated to be 2288K [9]. This is the highest possible temperature that can be attained at near-stoichiometric ratios for methane combustion and thus constitutes an upper limit for temperatures in numerical simulations. From fig. 5.1 it is evident that thermal NO production is significant only at temperatures above 1800K and at temperatures above 2200K the production is 50 times higher. The strong dependence of the rate of NO formation on peak temperatures and the dependence of the highest attainable temperature on the equivalence ratio thus is the key to NO reduction. The NO formation rate varies strongly with equivalence ratio. For instance the variation of the initial NO formation rate for adiabatic combustion of kerosene is shown in fig. 11.3 [33]. The effectiveness of two commonly used methods for NOx reduction 1.) variation of air to fuel equivalence ratio and 2.) flue gas re-circulation (FGR) are illustrated in the following subsections using the BFR test case.

11.2.1 Air to Fuel Equivalence Ratio Variation

From fig. 11.3 it is evident that diluting the combustion gases with excess air decreases the flame temperature for fuel-lean combustion and increases the temperature for fuel-rich combustion.

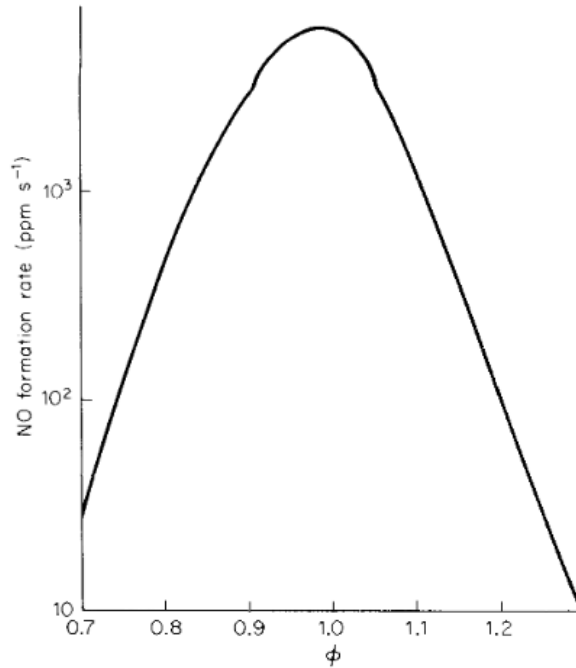


Figure 11.3: NO formation rate variation with equivalence ratio (ϕ) [33]

To illustrate the strong dependence of NO formation on equivalence ratio we do numerical experiments using the Burner Flow Reactor geometry. The simulations are performed at different equivalence ratio and the mass fractions of NO(in ppm) and CO(in promille) are calculated at the outlet. The 2-step reaction mechanism of Westbrook and Dryer (section 7.6) for methane combustion is used for all numerical experiments in this chapter. The global equivalence ratio for the BFR is calculated using eq. (10.2). The mass flow rate of the fuel at the primary fuel inlet and secondary fuel inlet are changed instead of changing the air inflow to achieve a particular equivalence ratio. The mass flow rates are changed by changing the velocity of fuel injection U_1 at the primary fuel inlet and U_2 at the secondary fuel inlet keeping the mass flow of air at the ambient air inlet the same for all the cases. The averages of temperature and mass fractions of NO and CO are calculated at the outlet using area weighted averages.

$$\text{Area Average: } \bar{\Phi} = \frac{\sum_i \Phi_i A_i}{\sum_i A_i}, \quad \text{Volume Average: } \bar{\Phi} = \frac{\sum_i \Phi_i V_i}{\sum_i V_i}$$

(U1, U2)	Eq. Ratio	Max. T(K)	Vol. Avg T(K)	Outlet Avg T(K)	NO_{ppm}	$CO_{promille}$
(0.787, 1.0)	0.4	2212.48	740.740	637.114	2.93	8.73
(2.053, 1.0094)	0.6	2202.51	826.201	665.704	5.78	6.03
(2.053, 1.7309)	0.8	2231.25	878.554	685.365	14.23	7.36
(2.053, 2.4520)	1.0	2284.97	932.106	716.467	23.84	26.55
(2.053, 3.1736)	1.2	2299.19	921.251	712.825	29.69	42.80
(2.053, 3.8950)	1.4	2221.74	919.726	710.711	3.56	50.02

ppm - parts per million.
promille - parts per thousand.
All quantities in SI units.

Table 11.2: Variation of NO and CO mass fractions with equivalence ratio (ϕ).

The variation of NO mass fraction with equivalence ratio is shown in fig. 11.4. The data set of 6 points is fit with a polynomial of degree 5 using Matlab. It is interesting to observe the sensitivity of NO formation with equivalence ratio. The maximum NO formation at equivalence ratio 1.2 is 10 times the NO formation at equivalence ratio 0.4. The contour plots of temperature and NO mass fraction are shown in fig. 11.5-11.8. The statistics of the temperature distribution in the domain for different equivalence ratios are represented as histograms in fig. 11.9. Since the NO formation is significant only at temperatures higher than 1800K (fig. 5.1), the numerical cells above 1800K are only considered in the computation of the statistics.

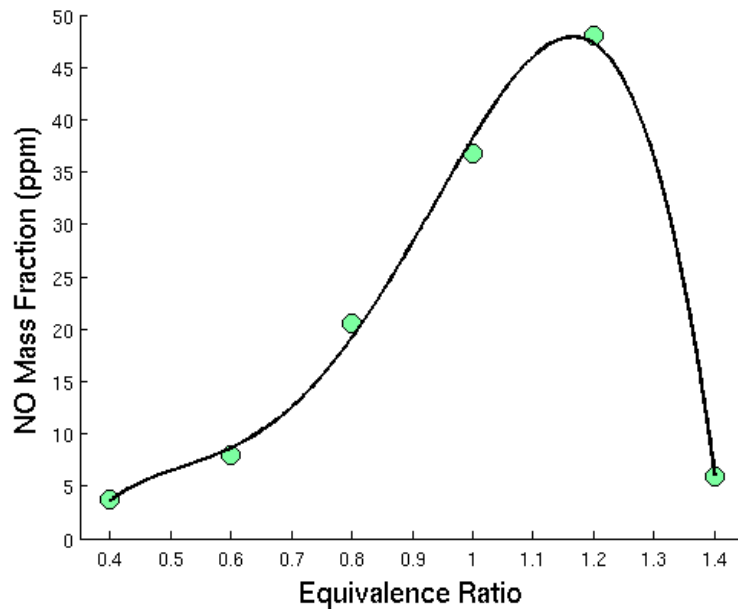


Figure 11.4: Variation of NO mass fraction with equivalence ratio.

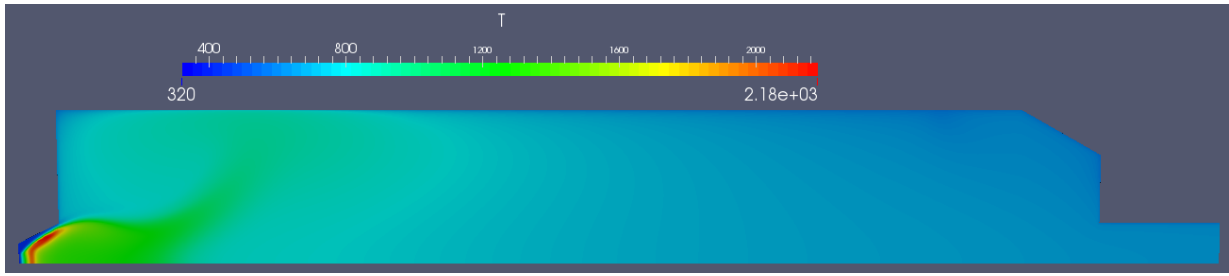


Figure 11.5: Contours of Temperature(K) for equivalence ratio 0.4

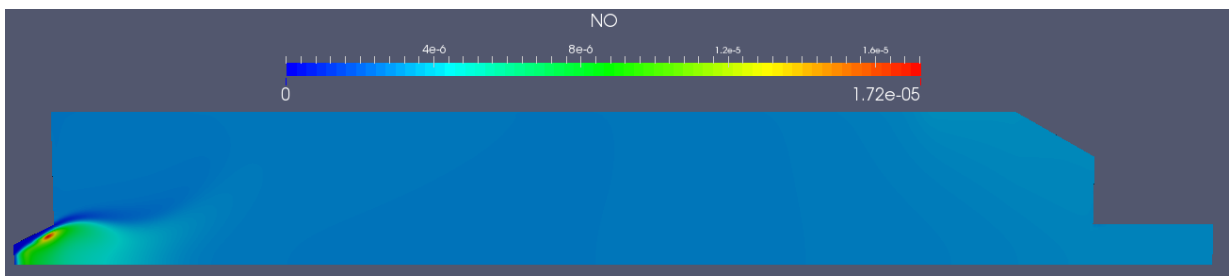


Figure 11.6: Contours of NO mass fraction for equivalence ratio 0.4

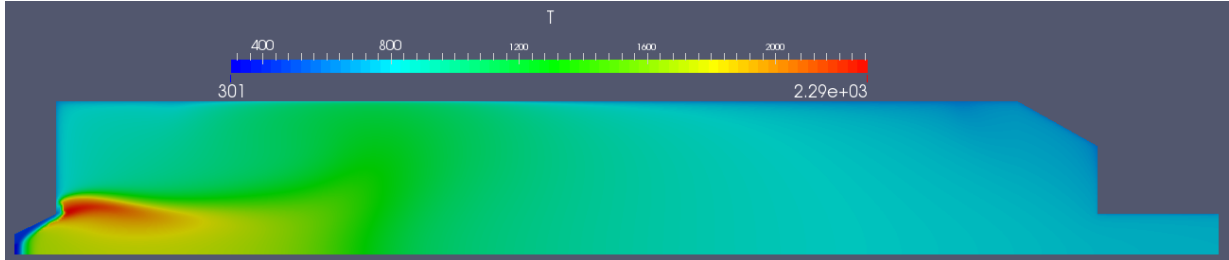


Figure 11.7: Contours of Temperature(K) for equivalence ratio 1.2

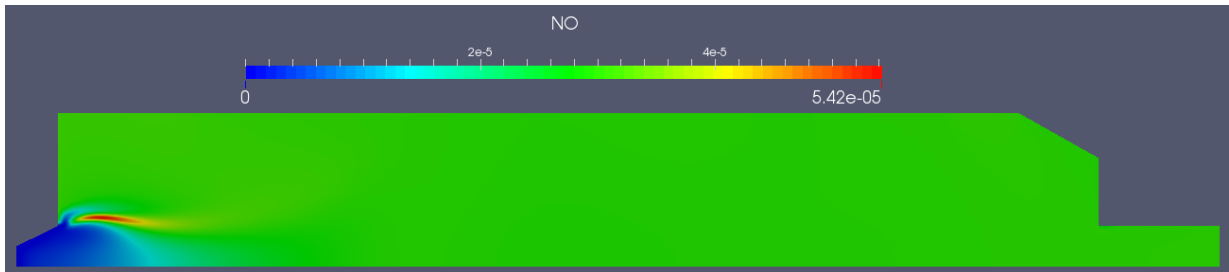


Figure 11.8: Contours of NO mass fraction for equivalence ratio 1.2

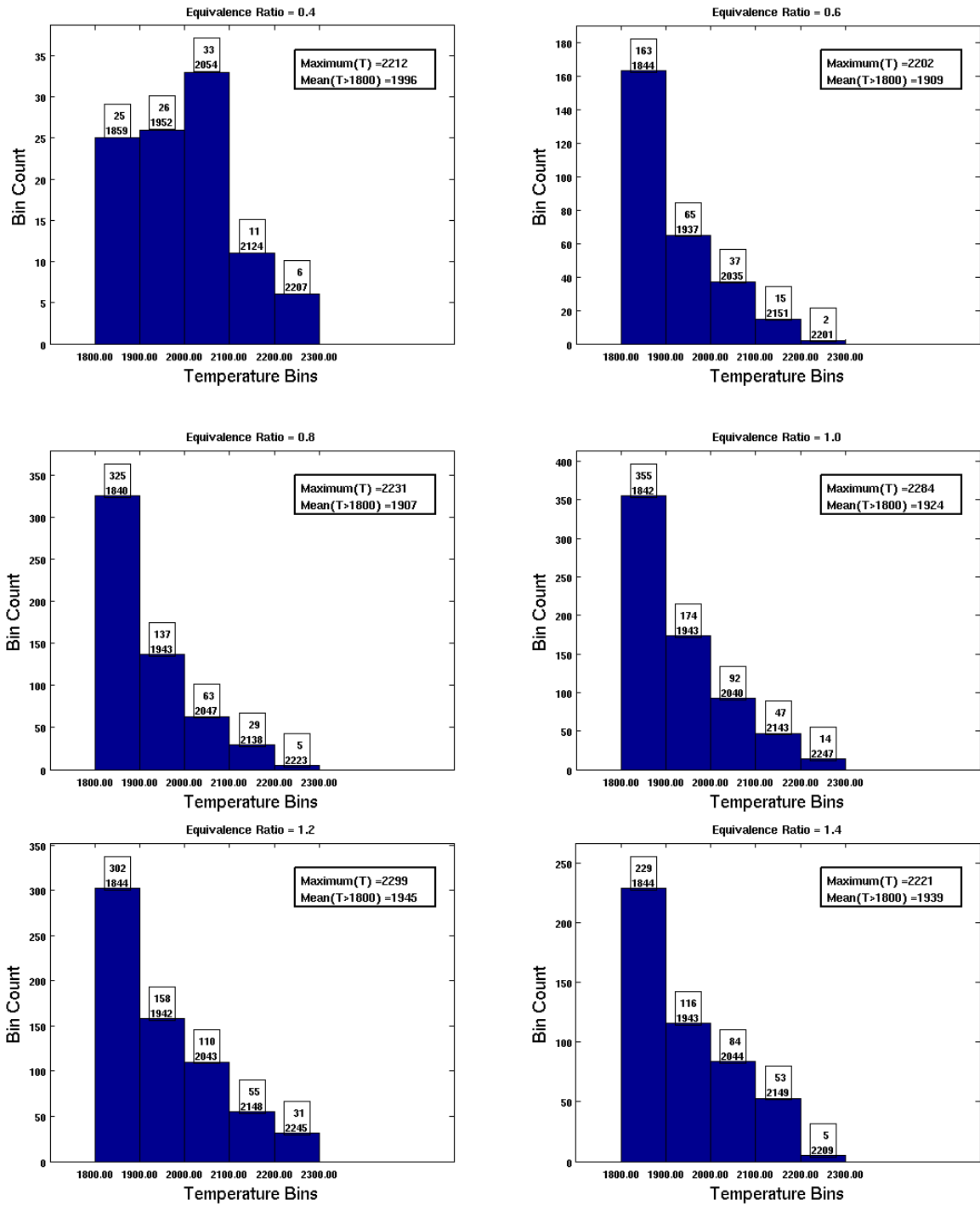


Figure 11.9: Temperature(K) distribution statistics at different equivalence ratios.

11.2.2 Flue Gas Re-circulation (FGR)

A commonly used method in industry is the re-circulation of the non-reactive flue gases from the furnace exhaust into the flame via duct work external to the furnace. Although the furnace or flue gases are hot they are considerably cooler than the flame itself. The re-circulated flue gases act as diluent, reducing the flame temperature which results into low NOx emissions [58]. This method reduces the adiabatic flame temperature and gives more effective control than changing the air to fuel equivalence ratio by altering the air/fuel inflow rate. Injection of other diluents such as steam can also be used to reduce the NO formation rates but at the cost of reduced system efficiency [33].

FGR reduces the NO formation in the following two ways [59].

- The recirculated flue gases mostly contain nitrogen and therefore are inert to the combustion process. When mixed with the combustion air, they decrease the average oxygen content thus starving the NOx-forming reactions of one of the required ingredients.
- The recirculated flue gases are relatively cooler than the hot gases in the flame zone, thereby acting as a heat sink, absorbing heat from the flame and lowering peak flame temperatures.

The flue gas re-circulation is achieved in OpenFOAM by using the area-weighted mass fractions at the outlet and prescribing them at the inlet using swak4Foam library. The mass fraction of a specie s is calculated at the inlet in the following way

$$Y_s^{inlet} = (Y_s^{air} + FGR * Y_s^{outlet}) / (1 + FGR)$$

where FGR is the percentage of the total flue gas flow that is routed back into the burner and mixed with ambient air. The FGR method is observed to be very effective in reducing the peak flame temperatures and thus NOx formation as seen in table 11.3. There is significant NOx reduction seen from 29.69 ppm to 0.008 ppm for a 10% flue gas re-circulation.

(U1, U2)	Eq. Ratio	Max. T(K)	Vol. Avg T(K)	Outlet Avg T(K)	NO _{ppm}	CO _{promille}
(2.053, 3.1736)	1.2	2299.19	921.251	712.825	29.69	42.80
(2.053, 3.1736)	FGR 10%	1867.83	864.064	674.413	0.008	36.39
(2.053, 3.1736)	FGR 20%	1691.03	848.967	666.167	0.0005	32.11

ppm - parts per million.

promille - parts per thousand.

All quantities in SI units.

Table 11.3: Variation of NO and CO mass fractions with flue gas re-circulation.

11.3 Summary

The new solver NOxFoam implemented in OpenFOAM for thermal NO prediction is compared with the NOx post-processing utility in ANSYS Fluent. The results are found to be in good agreement for the simple test case defined using the Burner Flow Reactor geometry. The NOxFoam utility is further used to demonstrate the effectiveness of the methods 1.) variation of air to fuel equivalence ratio and 2.) flue gas re-circulation (FGR) in reducing the NOx emissions using the Burner Flow Reactor test case. The methods are found to be very effective in reducing thermal NO emissions by altering the boundary conditions at the inlet for the same furnace geometry and inlet burner specifications.

Chapter 12

Conclusions and Recommendations

The main goal of this thesis was to assess the applicability of the OpenFOAM toolbox in modelling turbulent non-premixed combustion in complex industrial furnaces as an alternative to the costly commercial packages. In this chapter, the conclusions and recommendations based on the test cases and the problems encountered are addressed.

12.1 Conclusions

Several possibilities exist for simulating turbulent combustion from the widely used commercial software for complex industrial problems to customized in-house codes for relatively simple problems. In the present work the capabilities of the emerging open source toolbox OpenFOAM for turbulent combustion simulations were assessed using a benchmark problem and an industrial furnace test case.

To summarize the following tasks were accomplished in the present work and the findings are reported.

- The new solver implemented in OpenFOAM for turbulent combustion and radiation modelling `furnaceFoam` was evaluated using the benchmark test case Sandia Flame D. The simulation results were found to be in good agreement with the experimental data for Sandia Flame D indicating the adequacy and accuracy of the implemented solver `furnaceFoam` and its readiness for further combustion application development.
- The Burner Flow Reactor test case was used as an alternative to the Almatris Kiln for comparison of the simulation results in OpenFOAM with ANSYS Fluent. The relatively smaller size of the BFR as compared to the Almatris Kiln enabled carrying out numerical experiments for NO_x reduction in the available short time frame of the M.Sc thesis. The OpenFOAM results were found to be in good agreement with ANSYS Fluent except for some differences in the predicted temperature field near the inlet which is ascribed to the different combustion models used.

- Mesh non-orthogonality tests for the Burner Flow Reactor geometry were performed and it was concluded that OpenFOAM is sensitive to mesh non-orthogonality because of the way gradients are computed at the cell faces. It was observed that non-orthogonal corrections significantly reduce the numerical errors but cannot completely eliminate them. Therefore, a quad-dominant mesh in 2D and hex-dominant mesh in 3D with minimum mesh non-orthogonality is recommended
- A new solver in OpenFOAM for thermal NO prediction NOxFoam was developed. The solver was validated with the NOx post-processing utility in ANSYS Fluent using the Burner Flow Reactor geometry. The effectiveness of NOx reduction mechanisms including the variation of air to fuel equivalence ratio and flue gas re-circulation (FGR) was demonstrated using the Burner Flow Reactor test case.

Overall, the freely available OpenFOAM toolbox was found to be a good alternative to costly commercial packages in terms of accuracy and robustness. However, the simulation time for reaching the steady state in reacting flow simulations in OpenFOAM was found to be 10 folds higher than its commercial counterpart ANSYS Fluent. This puts a restriction on the use of OpenFOAM for more complex and larger industrial furnaces such as the Almatris Kiln. Therefore, the development of a steady state combustion solver based on equilibrium chemistry models in OpenFOAM is indispensable if larger industrial furnaces such as the Almatris Kiln are to be simulated in realizable time.

12.2 Recommendations

This section lists the further developments required for making OpenFOAM a good and likely replacement for expensive commercial packages for combustion applications.

- In chapter 4 the external wall heat flux boundary condition to realize the conjugate heat transfer between the hot gases in the interior of the furnace and the furnace insulation walls was described. The externalWallHeatFlux boundary condition in OpenFOAM can be used as a crude approximation to model the conjugate heat transfer problem. However, a more comprehensive approach would be to implement conjugate heat transfer in the combustion solver using the existing solver chtMultiRegionFoam. The chtMultiRegionFoam solver requires a compressible solver for the fluid region and therefore cannot be directly used for implementation and would need further development. This can be accomplished as a part of the future work.
- In future work it would be of great interest to develop the missing equilibrium chemistry models in OpenFOAM. The widely used eddy break-up equilibrium chemistry model for turbulent diffusion flames is of particular interest. To assess the applicability of the model, the calculation of the Damkohler number can be integrated into the combustion solver.

- To reduce the calculation time, steady state gas-combustion solvers in OpenFOAM needs to be developed and validated. An alternative to creating a new steady state gas-combustion solver is to speed-up the transient calculations using the already available local time stepping (LTS) version of the reactingFoam solver called LTSReactingFoam [60]. LTSReactingFoam can further be used instead of reactingFoam to derive the new solver for turbulent combustion that includes radiation modelling.
- To predict total NO formation from all sources, the computation of “prompt NO” and “fuel NO” needs to be integrated into the new solver NOxFoam for “thermal NO” prediction. The solver can further be validated using experimental data.
- The development of OpenFOAM as a CFD toolbox began with much emphasis on the implementation of physical models with little emphasis on efficient numerics and linear algebra sub-routines for solving the resulting large sparse linear systems after finite volume discretisation. OpenFOAM has its own implementation of linear solvers based on the LDU matrix class and provides its users with the following linear solvers PBiCG, PCG, GAMG and an array of preconditioners [35]. Profiling OpenFOAM proved the linear solvers with sparse matrix vector multiplication to be its most computationally intensive kernel [61]. There is a lot of scope for further development of the linear solvers and numerical linear algebra libraries in OpenFOAM. A few recommendations are listed here
 - At present OpenFOAM employs the classical segregated approach where the partial differential equations are solved one at a time, with the inter-equation coupling treated in the explicit manner. Non-linear equations are first linearised before the discretisation and the non-linear terms are lagged. The SIMPLE algorithm of Patankar, S. V. and Spalding, D.B. (1972) [46] is a classic example following this approach. The SIMPLE like algorithms segregates the solution of the pressure and velocity field, in many cases resulting in slow convergence.

An alternative to the segregated approach is to couple the continuity and momentum equations while keeping the model equations for other physical quantities such as turbulence, chemical species and enthalpy transport segregated. The coupled Navier–Stokes system can then be solved with a Krylov subspace method with a preconditioner. For steady-state simulations the use of SIMPLE as a solver can then be replaced by the use of SIMPLE as a preconditioner as demonstrated in [62]. It was further demonstrated in [62] that compared to the classical SIMPLE solver, SIMPLER preconditioning reduces both the number of nonlinear iterations and CPU time by a significant factor. This can be implemented in OpenFOAM beginning from the isothermal flow solvers and further extending it to the more complex reacting flow solvers.

- The naive implementation of OpenFOAM linear solvers can be replaced by the efficient sparse linear solvers available from the PETSc suite of data structures and routines for the scalable (parallel) solution of scientific applications.

- The Induced Dimension Reduction (IDR(s)) method [63] which requires considerably less matrix-vector multiplications than BiCG and BiCGSTAB is a promising method and can be included in the suite of linear solvers available with OpenFOAM.
- Offloading parts of computations specifically, solution of linear systems to a graphics accelerator (GPU) [64] is another alternative which can result into significant speed-up of the OpenFOAM applications.

As a closing remark the OpenFOAM toolbox now is due to the immense contributions of the open source community in the form of research articles, training presentations and thesis reports such as this which are alternatives to a missing detailed documentation. An important recommendation to the reader is to be aware of the latest developments in OpenFOAM by following CFD Online where most of the OpenFOAM issues are discussed and resolved. The OpenFOAM users at the same time have a responsibility of updating the OpenFOAM user community with their developments.

Bibliography

- [1] Dominik Christ Bjorn Lilleberg. “Numerical Simulation with an Extinction Database for Use with the Eddy Dissipation Concept for Turbulent Combustion”. In: *Flow, Turbulence and Combustion* 91.2 (Sept. 2013), pp. 319–346.
- [2] Kjell Erik Rian Dmitry A. Lysenko Ivar S. Ertesv. “Numerical Simulation of Non-premixed Turbulent Combustion Using the Eddy Dissipation Concept and Comparing with the Steady Laminar Flamelet Model”. In: *Flow, Turbulence and Combustion* 93.4 (Dec. 2014), pp. 577–605.
- [3] Kjell Erik Rian Dmitry A. Lysenko Ivar S. Ertesv. “Numerical Simulations of the Sandia Flame D Using the Eddy Dissipation Concept”. In: *Flow, Turbulence and Combustion* 93.4 (Dec. 2014), pp. 665–687.
- [4] J.H. Frank R.S. Barlow. “Effects of turbulence on species mass fractions in methane/air jet flames”. In: *Symposium (International) on Combustion* 27.1 (1998), pp. 1087–1095.
- [5] Lasse Rosendahl Chungen YIN Soren Knudsen and Soren Lovmand. “Modeling of pulverised coal and biomass co-firing in a 150 KW swirling-stabilised burner and experimental validation”. In: *Proceedings of the International Conference on Power Engineering-09*. Nov. 2009.
- [6] Akwasi Boateng. *Rotary Kilns Transport Phenomena and Transport Processes*. Butterworth Heinemann, 2008.
- [7] Michele Pisaroni, R Sadi, and D Lahaye. “Counteracting ring formation in rotary kilns”. In: *Journal of Mathematics in Industry* 2.1 (2012), pp. 1–19.
- [8] Niels E. L. Nielsen Christian Andersen. “Numerical Investigation of a BFR using OpenFOAM”. MA thesis. Aalborg University, 2008.
- [9] H.K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. 2nd edition. Pearson Education Limited, 2007.
- [10] Norbert Peters. *Turbulent Combustion, Cambridge Monographs on Mechanics*. Cambridge University Press, 2014.
- [11] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, Oct. 2000.

- [12] François G. Schmitt. “About Boussinesq’s turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity”. In: *Comptes Rendus Mécanique* 335.9–10 (2007). Joseph Boussinesq, a Scientist of bygone days and present times, pp. 617–627. ISSN: 1631-0721. DOI: <http://dx.doi.org/10.1016/j.crme.2007.08.004>. URL: <http://www.sciencedirect.com/science/article/pii/S1631072107001386>.
- [13] Ansys Fluent. *Fluent 6.3 users guide*. URL: https://www.sharcnet.ca/Software/Fluent6/html/ug/main_pre.htm.
- [14] Eric Furbo. “Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers.” MA thesis. Uppsala University, 2010.
- [15] “Copyright”. In: *Combustion (Fourth Edition)*. Ed. by Irvin Glassman and Richard A. Yetter. Fourth Edition. Burlington: Academic Press, 2008, pp. iv–. ISBN: 978-0-12-088573-2. DOI: <http://dx.doi.org/10.1016/B978-0-12-088573-2.00025-7>. URL: <http://www.sciencedirect.com/science/article/pii/B9780120885732000257>.
- [16] *OpenFOAM Tutorial reactingFoam: Simple Gas Phase Reaction*. URL: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2007/AndreasLundstrom/reactingFoam.pdf.
- [17] Lin Wang et al. “Comparison of Different Global Combustion Mechanisms Under Hot and Diluted Oxidation Conditions”. In: *Combustion Science and Technology* 184.2 (2012), pp. 259–276. DOI: 10.1080/00102202.2011.635612. eprint: <http://dx.doi.org/10.1080/00102202.2011.635612>. URL: <http://dx.doi.org/10.1080/00102202.2011.635612>.
- [18] *GRI-Mech 3.0*. URL: <http://combustion.berkeley.edu/gri-mech/version30/text30.html>.
- [19] *NIST-JANAF Thermochemical Tables*. URL: <http://kinetics.nist.gov/janaf/>.
- [20] Epaminodas Mastorakos Tarek Echehki. *Turbulent Combustion Modeling: Advances, New Trends and Perspectives*. 1st ed. Vol. 95. Springer Netherlands, 2011.
- [21] Dominik Christ. *Simulating the combustion of gaseous fuels, 6th OpenFoam Workshop Training Session, June 2011*. URL: https://openfoamwiki.net/images/8/8a/Christ_OFW6_slides.pdf.
- [22] Bjorn F Magnussen. “The Eddy Dissipation Concept—A Bridge Between Science and Technology”. In: *ECCOMAS thematic conference on computational combustion*. 2005, pp. 21–24.
- [23] *Combustion*. URL: http://www.cfd-online.com/Wiki/Combustion#The_Damk.C3.B6hler_Number.
- [24] *KALMAN KNIZHNIK - DERIVATION OF THE STEFAN-BOLTZMANN LAW*. URL: http://www.pha.jhu.edu/~kknizhni/StatMech/Derivation_of_Stefan-Boltzmann_Law.pdf.

- [25] “Radiative Heat Transfer, Chapter 4”. In: *ERCRAFTAC Best Practice Guide on CFD of combustion*. Ed. by M. Mancini P.J. Coelho and D.J.E.M. Roekaerts. ERCRAFTAC, 2015.
- [26] Michael F. Modest. *Radiative Heat Transfer*. Academic Press, Burlington, 2003.
- [27] Tarit K Bose. *High Temperature Gas Dynamics: An Introduction for Physicists and Engineers*. 2nd ed. Springer International Publishing, 2014.
- [28] Eddy Petro Matthieu Melot Ricardo Camarero. “Comparison of Two Models for Radiative Heat Transfer in High Temperature Thermal Plasmas”. In: *Modelling and Simulation in Engineering 2011* (2011). DOI: 10.1155/2011/285108. URL: <http://dx.doi.org/10.1155/2011/285108>.
- [29] M.A. Founti E.P. Keramida H.H. Liakos. “Radiative heat transfer in natural gas-fired furnaces”. In: *International Journal of Heat and Mass Transfer* 43.10 (2000), pp. 1801–1809. DOI: 10.1016/S0017-9310(99)00244-6. URL: <http://www.sciencedirect.com/science/article/pii/S0017931099002446>.
- [30] *Conjugate Heat Transfer*. URL: <http://www.comsol.com/blogs/conjugate-heat-transfer/>.
- [31] S. Mostafa Ghiaasiaan. Cambridge University Press, 2011. ISBN: 978-1-107-00350-7. URL: <http://app.knovel.com/hotlink/toc/id:kpCHMT000J/convective-heat-mass/convective-heat-mass>.
- [32] *Natural convection on cylinders and spheres*. URL: http://www.thermalfluidscentral.org/encyclopedia/index.php/Natural_convection_on_cylinders_and_spheres.
- [33] Richard C. Flagan and J.H. Seinfeld. *Fundamentals of air pollution engineering*. Prentice Hall, 1988. ISBN: 9780133325379. URL: <http://authors.library.caltech.edu/25069/>.
- [34] TJ Foster et al. “Measurement and prediction of NO and NO₂ emissions from aero engines”. In: *RTO Meeting proceedings*. 1999.
- [35] CFD Direct. *OpenFOAM user guide 2.3.1*. URL: <http://cfd.direct/openfoam/user-guide/>.
- [36] Open FOAM Wiki. *People and institutions that use OpenFOAM*. URL: https://openfoamwiki.net/index.php/Main_Users.
- [37] Ed Komen et al. “Quasi-DNS capabilities of OpenFOAM for different mesh types”. In: *Computers & Fluids* 96 (2014), pp. 87–104.
- [38] Dr. SK Ajmani Dr.ir Eelco van Vliet Ir. Dirk van der Plas. *Development of an OpenFOAM CFD model of the electromagnetically stirred steel flow in the mould of a billet caster, 5th OpenFOAM Workshop, Chalmers, Gothenberg, Sweden, June 21-24, 2010*. URL: <http://web.student.chalmers.se/groups/ofw5/Abstracts/EelcoVanVlietAbstractOFW5.pdf>.

- [39] *Introductory OpenFOAM Course: Tips and tricks in OpenFOAM*. URL: <http://www.dicat.unige.it/guerrero/of2015a/9tipsandtricks.pdf>.
- [40] Z Warhaft. “Passive scalars in turbulent flows”. In: *Annual Review of Fluid Mechanics* 32.1 (2000), pp. 203–240.
- [41] Hrvoje Jasak. “Error analysis and estimation for the finite volume method with applications to fluid flows”. PhD thesis. Imperial College London (University of London), 1996.
- [42] *Discretisation of the diffusion term*. URL: http://www.cfd-online.com/Wiki/Diffusion_term.
- [43] Guilherme Coelho Cunha Antonio de Oliveira Samel Moraes Paulo Laranjeira da Cunha Lage. “ANALYSIS OF THE NON-ORTHOGONALITY CORRECTION OF FINITE VOLUME DISCRETIZATION ON UNSTRUCTURED MESHES”. In: 2013. URL: <http://www.abcm.org.br/anais/cobem/2013/PDF/740.pdf>.
- [44] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Berlin, 1999. ISBN: 9783540653738.
- [45] H. G. Weller et al. “A Tensorial Approach to Computational Continuum Mechanics Using Object-oriented Techniques”. In: *Comput. Phys.* 12.6 (Nov. 1998), pp. 620–631. ISSN: 0894-1866. DOI: 10.1063/1.168744. URL: <http://dx.doi.org/10.1063/1.168744>.
- [46] S.V Patankar and D.B Spalding. “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”. In: *International Journal of Heat and Mass Transfer* 15.10 (1972), pp. 1787–1806.
- [47] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. Series in computational methods in mechanics and thermal sciences. Taylor & Francis, 1980. ISBN: 9780891165224.
- [48] *OpenFOAM guide - The PIMPLE algorithm in OpenFOAM*. URL: http://openfoamwiki.net/index.php/OpenFOAM_guide/The_PIMPLE_algorithm_in_OpenFOAM.
- [49] “Turbulent Combustion Modeling, Chapter 7”. In: URL: <http://www.tfd.chalmers.se/~valeri/book-chapt7.pdf>.
- [50] G D’Errico, D Ettorre, and T Lucchini. *Comparison of combustion and pollutant emission models for DI diesel engines*. Tech. rep. SAE Technical Paper, 2007.
- [51] Fabian Peng Karrholm. “Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion”. PhD thesis. Department of Applied Mechanics, Chalmers University of Technology, 2008. URL: <http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/>.
- [52] Robert Barlow. *Sandia/TUD Piloted CH₄/Air Jet Flames*. Tech. rep. Includes scalar data for flames C,D,E, and F. International Workshop, Measurement and Computation of Turbulent Non Premixed Flames, Jan. 2003. URL: <http://www.sandia.gov/TNF/DataArch/FlameD.html>.

- [53] *ERCOFTAC QNET-CFD Wiki Forums*. URL: http://qnet-ercoftac.cfms.org.uk/w/index.php/Description_AC2-09.
- [54] *Introductory OpenFOAM Course: Meshing and mesh quality assessment for CFD computations*. URL: http://www.dicat.unige.it/guerrero/of2015a/3_2mesh_quality_assessment.pdf.
- [55] *OpenFOAM Workshop 2014: Effects of grid quality on solution accuracy*. URL: <https://afinemesh.files.wordpress.com/2014/07/ofw20141.pdf>.
- [56] CHARLES K. WESTBROOK and FREDERICK L DRYER. “Simplified Reaction Mechanisms for the Oxidation of Hydrocarbon Fuels in Flames”. In: *Combustion Science and Technology* 27.1-2 (1981), pp. 31–43. DOI: 10.1080/00102208108946970. eprint: <http://dx.doi.org/10.1080/00102208108946970>. URL: <http://dx.doi.org/10.1080/00102208108946970>.
- [57] *Contrib/PyFoam*. URL: <https://openfoamwiki.net/index.php/Contrib/PyFoam>.
- [58] C.E. Baukal. *Industrial Combustion Pollution and Control*. Environmental Science & Pollution. CRC Press, 2003. ISBN: 9780203912782.
- [59] S. Londerville and C.E. Baukal. *The Coen & Hamworthy Combustion Handbook: Fundamentals for Power, Marine & Industrial Applications*. Industrial Combustion. Taylor & Francis, 2013. ISBN: 9781439873342.
- [60] Kar Mun Pang et al. “Development and validation of a local time stepping-based PaSR solver for combustion and radiation modeling”. In: *Proceedings of 8th International OpenFOAM Workshop*. 2014.
- [61] Mottaqiallah Taouil. “A Hardware Accelerator for the OpenFoam Sparse Matrix-Vector Product”. MA thesis. Delft University of Technology, 2009. URL: http://ce-publications.et.tudelft.nl/publications/347_a_hardware_accelerator_for_the_openfoam_sparse_matrixvector.pdf.
- [62] CM Klaij and C Vuik. “SIMPLE-type preconditioners for cell-centered, colocated finite volume discretization of incompressible Reynolds-averaged Navier–Stokes equations”. In: *International Journal for Numerical Methods in Fluids* 71.7 (2013), pp. 830–849.
- [63] Martin B van Gijzen and Peter Sonneveld. “The IDR (s) method for solving non-symmetric systems: a performance study for CFD problems”. In: *High Performance Algorithms for Computational Science and Their Applications* (2008).
- [64] Amani AlOnazi et al. “Design and optimization of openfoam-based cfd applications for hybrid and heterogeneous hpc platforms”. In: *arXiv preprint arXiv:1505.07630* (2015).

Appendix A

simpleFoam source code

```
1 /* ----- simpleFoam source code */
2
3 #include "fvCFD.H"
4 #include "singlePhaseTransportModel.H"
5 #include "RASModel.H"
6 #include "simpleControl.H"
7 #include "fvIOoptionList.H"
8
9 // * * * * *
10
11 int main(int argc, char *argv[])
12 {
13     // 1. Pre-processing
14     #include "setRootCase.H"
15     #include "createTime.H"
16     #include "createMesh.H"
17     #include "createFields.H"
18     #include "createFvOptions.H"
19     #include "initContinuityErrs.H"
20
21     simpleControl simple(mesh);
22
23     // * * * * *
24
25     Info<< "\nStarting time loop\n" << endl;
26
27     // SIMPLE loop
28     while (simple.loop())
29     {
30         Info<< "Time==" << runTime.timeName() << nl << endl;
31
32         {
33             // 2. Momentum Predictor
34             /* The momentum equation is solved first.
35              The exact pressure gradient source term is
36              not known at this stage the pressure field
```

```

37         from the previous time-step is used instead.
38         This stage is called the momentum predictor.
39         The solution of the momentum equation gives
40         an approximation of the new velocity field.
41     */
42     // #include "uEqn.H"
43     {
44         /* Solve the discretized momentum equation
45         to compute the intermediate velocity field. */
46
47         // Define the equation for U.
48         tmp<fvVectorMatrix> UEqn
49         (
50             fvm::div(phi, U)
51             + turbulence->divDevReff(U)
52             ==
53             fvOptions(U)
54         );
55
56         // Under-relax the equation for U.
57         UEqn().relax();
58         fvOptions.constrain(UEqn());
59
60         // Solve using previous p.
61         solve(UEqn() == -fvc::grad(p));
62         fvOptions.correct(U);
63     }
64
65
66     // #include "pEqn.H"
67     {
68         volScalarField rAU(1.0/UEqn().A());
69         volVectorField HbyA("HbyA", U);
70         HbyA = rAU*UEqn().H();
71         UEqn.clear();
72
73         // 3. Compute the mass fluxes at the cell faces.
74         surfaceScalarField phiHbyA("phiHbyA", fvc::interpolate(
75             HbyA) & mesh.Sf());
76         fvOptions.makeRelative(phiHbyA);
77         adjustPhi(phiHbyA, U, p);
78
79         /* Define and solve the pressure equation using the
80         predicted velocities and repeat for the prescribed
81         number of non-orthogonal corrector steps. */
82         while (simple.correctNonOrthogonal())
83         {
84             // 4. Pressure Correction
85             fvScalarMatrix pEqn
86             (

```

```

87         fvm::laplacian(rAU, p) == fvc::div(phiHbyA)
88     );
89     pEqn.setReference(pRefCell, pRefValue);
90     pEqn.solve();
91
92     // 5. Correct the mass fluxes at the cell faces.
93     if (simple.finalNonOrthogonalIter())
94     {
95         phi = phiHbyA - pEqn.flux();
96     }
97 }
98
99 // Calculate continuity errors
100 #include "continuityErrs.H"
101
102 /* 6. Correct the velocities on the basis of
103    the new pressure field. */
104 // Explicitly under-relax the pressure for
105 // momentum corrector.
106 p.relax();
107 U = HbyA - rAU*fvc::grad(p);
108
109 // 7. Update the boundary conditions.
110 U.correctBoundaryConditions();
111 fvOptions.correct(U);
112 }
113
114
115 }
116 // 8. Correct for turbulence.
117 turbulence->correct();
118
119 runTime.write();
120
121 Info<< "ExecutionTime===" << runTime.elapsedCpuTime() << "s"
122     << "==" << runTime.elapsedClockTime() << "s"
123     << nl << endl;
124 }
125
126 Info<< "End\n" << endl;
127
128 return 0;
129 }
130
131
132 // *****
133 //

```


Appendix B

furnaceFoam source code

```
1 /* ----- furnaceFoam source code */
2
3 #include "fvCFD.H"
4 #include "turbulenceModel.H"
5 #include "psiCombustionModel.H"
6 #include "multivariateScheme.H"
7 #include "pimpleControl.H"
8 #include "fvIOoptionList.H"
9 #include "radiationModel.H"
10
11 int main(int argc, char *argv[])
12 {
13     // 1. Pre-processing
14     #include "setRootCase.H"
15     #include "createTime.H"
16     #include "createMesh.H"
17     #include "readGravitationalAcceleration.H"
18     #include "createFields.H"
19
20     #include "createFvOptions.H"
21
22     #include "createRadiationModel.H"
23
24     #include "initContinuityErrs.H"
25     #include "readTimeControls.H"
26     #include "compressibleCourantNo.H"
27     #include "setInitialDeltaT.H"
28
29     pimpleControl pimple(mesh);
30
31     // * * * * *
32     //
33     Info<< "\nStarting time loop\n" << endl;
34
35     while (runTime.run()) // to iterate in time.
```

```

36 {
37 // 2. Calculate T
38 {
39 // Read the control parameters used by setDeltaT
40 #include "readTimeControls.H"
41
42 // Calculates and outputs the mean and maximum Courant Numbers.
43 #include "compressibleCourantNo.H"
44
45 /* Reset the timestep to maintain a constant maximum courant Number
46 .
47 Reduction of time-step is immediate, but increase is damped to
48 avoid
49 unstable oscillations. */
50 #include "setDeltaT.H"
51
52 runTime++;
53 Info<< "Time===" << runTime.timeName() << nl << endl;
54 }
55 // 3. Solve the continuity equation to compute density.
56 #include "rhoEqn.H"
57 while (pimple.loop()) // outer iteration nOuterCorrectors = 1 (= 1
58 // means PISO mode)
59 {
60 // 4. Momentum Predictor
61 // #include "UEqn.H"
62 // * * * * *
63 // * * * * //
64 fvVectorMatrix UEqn
65 (
66     fvm::ddt(rho, U)
67     + fvm::div(phi, U)
68     + turbulence->divDevRhoReff(U)
69     ==
70     rho*g
71     + fvOptions(rho, U)
72 );
73 UEqn.relax();
74 fvOptions.constrain(UEqn);
75
76 // if momentum predictor is turned off, U from previous time-
77 // step is used,
78 // if it is on once the solve method is executed U is updated
79 // using the old
80 // pressure field.
81 if (pimple.momentumPredictor()) // set No in PIMPLE control
82 {

```

```

81         solve(UEqn == -fvc::grad(p));
82
83         fvOptions.correct(U);
84         K = 0.5*magSqr(U);
85     }
86     // * * * * *
87     // * * * * //
88
89     // 5. Species Transport
90     // #include "YEqn.H"
91     // * * * * *
92     // * * * * //
93     tmp<fv::convectionScheme<scalar>> mvConvection
94     (
95         fv::convectionScheme<scalar>::New
96         (
97             mesh,
98             fields,
99             phi,
100            mesh.divScheme("div(phi, Yi_h)")
101        );
102
103     // Solve transport equation for individual species mass
104     // fractions
105     {
106         reaction->correct();
107         dQ = reaction->dQ();
108         label inertIndex = -1;
109         volScalarField Yt(0.0*Y[0]);
110
111         forAll(Y, i)
112         {
113             if (Y[i].name() != inertSpecie)
114             {
115                 volScalarField& Yi = Y[i];
116
117                 fvScalarMatrix YiEqn
118                 (
119                     fvm::ddt(rho, Yi)
120                     + mvConvection->fvmDiv(phi, Yi)
121                     - fvm::laplacian(turbulence->muEff(), Yi)
122                     ==
123                     reaction->R(Yi)
124                     + fvOptions(rho, Yi)
125                 );
126
127                 YiEqn.relax();
128
129                 fvOptions.constrain(YiEqn);

```

```

129
130         YiEqn.solve(mesh.solver("Yi"));
131
132         fvOptions.correct(Yi);
133
134         Yi.max(0.0);
135         Yt += Yi;
136     }
137     else
138     {
139         inertIndex = i;
140     }
141 }
142
143     Y[inertIndex] = scalar(1) - Yt;
144     Y[inertIndex].max(0.0);
145 }
146 // * * * * *
147 // * * * * //
148 // 6. Energy Transport
149 // #include "EEqn.H"
150 // * * * * *
151 // * * * * //
152 // Solve energy equation to obtain new temperature field
153 {
154     volScalarField& he = thermo.he();
155
156     fvScalarMatrix EEqn
157     (
158         fvm::ddt(rho, he) + mvConvection->fvmDiv(phi, he)
159         + fvc::ddt(rho, K) + fvc::div(phi, K)
160         + (
161             he.name() == "e"
162             ? fvc::div
163             (
164                 fvc::absolute(phi/fvc::interpolate(rho), U),
165                 p,
166                 "div(phi,p)"
167             )
168             : -dpdt
169         )
170         - fvm::laplacian(turbulence->alphaEff(), he)
171         ==
172         reaction->Sh() // Source term for combustion
173         + radiation->Sh(thermo) // Source term for radiation
174         + fvOptions(rho, he)
175     );
176     EEqn.relax();
177

```

```

178         fvOptions.constrain(EEqn);
179
180         EEqn.solve();
181
182         fvOptions.correct(he);
183
184         // 7. Radiation Transport
185         // Solve transport equation for G
186         radiation->correct();
187
188         // 8. Update thermophysical properties
189         thermo.correct();
190
191         Info<< "min/max(T) = "
192             << min(T).value() << ", " << max(T).value() << endl;
193     }
194     // * * * * *
195     // * * * * *
196
197     // — Pressure corrector loop
198     while (pimple.correct()) // - inner iteration nCorrectors = 2
199     {
200         #include "pEqn.H"
201     }
202
203     // Its function is to determine whether to correct turbulence at
204     // every outer
205     // iteration or only at the last iteration.
206     // turbOnFinalIterOnly is set by default to true
207     if (pimple.turbCorr())
208     {
209         turbulence->correct();
210     }
211 } // Done PIMPLE - outer iteration
212
213     runTime.write();
214
215     Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
216         << "ClockTime = " << runTime.elapsedClockTime() << " s"
217         << nl << endl;
218 } // Done runTime.run()
219
220     Info<< "End\n" << endl;
221
222     return 0;
223 } // Done - main

```


Appendix C

System Properties

CPU architecture information of MPISSOR from sysfs and /proc/cpuinfo using lscpu.

Information	#
Architecture:	x86_64
CPU op-mode(s):	32-bit, 64-bit
Byte Order:	Little Endian
CPU(s):	4
On-line CPU(s) list:	0-3
Thread(s) per core:	1
Core(s) per socket:	4
Socket(s):	1
NUMA node(s):	1
Vendor ID:	GenuineIntel
CPU family:	6
Model:	26
Model name:	Intel(R) Xeon(R) CPU W3530
Stepping:	5
CPU MHz:	2800.141
BogoMIPS:	5600.28
Virtualization:	VT-x
L1d cache:	32K
L1i cache:	32K
L2 cache:	256K
L3 cache:	8192K
NUMA node0 CPU(s):	0-3

Table C.1: CPU architecture information of MPISSOR