A NON-ITERATIVE PRESSURE BASED ALGORITHM FOR THE COMPUTATION OF REACTING RADIATING FLOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET BİLGE UYGUR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CHEMICAL ENGINEERING

MARCH 2007

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Nurcan Baç
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. İ. Hakkı Tuncer
Co-Supervisor

Prof. Dr. Nevin Selçuk
Supervisor

Examining Committee Members

Prof. Dr. Haluk Aksel	(METU, ME)	
Prof. Dr. Nevin Selçuk	(METU, CHE)	
Prof. Dr. İ. Hakkı Tuncer	(METU, AE)	
Prof. Dr. Timur Doğu	(METU, CHE)	
Asst. Prof. Dr. Nimeti Döner	(DPU, ME)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ahmet Bilge Uygur

Signature :

ABSTRACT

A NON-ITERATIVE PRESSURE BASED ALGORITHM FOR THE COMPUTATION OF REACTING RADIATING FLOWS

Uygur, Ahmet Bilge

Ph.D., Department of Chemical Engineering

Supervisor: Prof. Dr. Nevin Selçuk

Co-Supervisor: Prof. Dr. İ. Hakkı Tuncer

March 2007, 181 pages

A non-iterative pressure based algorithm which consists of splitting the solution of momentum energy and species equations into a sequence of predictor-corrector stages was developed for the simulation of transient, reacting, radiating flows. A semi-discrete approach called the Method of Lines (MOL) which enables implicit time-integration at all splitting stages was used for the solution of conservation equations. The solution of elliptic pressure equation for the determination of pressure field was performed by a multi-grid solver (MUDPACK package). Radiation calculations were carried out by coupling previously developed gray and non-gray radiation models with the algorithm. A first order (global) reaction mechanism was employed to account for the chemistry.

The predictions of the algorithm for the following test cases: *i)* non-isothermal turbulent pipe flow and *ii)* laminar methane-air diffusion flame; were benchmarked against experimental data and numerical solutions available in the literature and

the capability of the code to predict transient solutions was demonstrated on these test cases. Favorable agreements were obtained for both test cases. The effect of radiation and non-gray treatment of the radiative properties were investigated on the second test case. It was found that incorporation of radiation has significant effect on temeprature and velocity fields but its effect is limited in species predictions. Executions with both radiation models revealed that the non-gray radiation model considered in the present study produces similar results with the gray model at a considerably higher computational cost. The algorithm developed was found to be an efficient and versatile tool for the time-dependent simulation of different flow scenarios constitutes the initial steps towards the computation of transient turbulent combustion.

Keywords: Non-iterative algorithms, Pressure based methods, Operator-splitting, Reacting radiating flows, Method of Lines.

ÖZ

TEPKİMELİ VE ISIL IŞIMALI AKIŞLARIN HESAPLANMASI İÇİN TEKRARSIZ BASINCA DAYALI ALGORİTMA

Uygur, Ahmet Bilge

Doktora, Kimya Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Nevin Selçuk

Ortak Tez Yöneticisi: Prof. Dr. İ. Hakkı Tuncer

Mart 2007, 181 sayfa

Zamana bağlı, tepkimeli ve ısıl ışımalı akışların sayısal benzetişimi için tekrarsız basınca dayalı ve momentum, enerji ve kütle denklemlerinin çözümünü tahmin etme-düzeltme aşamalarına ayıran bir algoritma geliştirilmiştir. Korunurluk denklemlerinin çözülmesinde, dolaylı zaman integrallenmesini her ayırma aşmasında sağlayan ve çizgiler metodu olarak da adlandırılan yarı-ayrık bir yaklaşım kullanılmıştır. Basınç alanının belirlenmesi için çözülmesi gereken elliptik basınç denklemi çoklu-nokta çözücüsü (MUDPACK paketi) ile sağlanmıştır. Isıl ışınım hesaplamaları, daha önce geliştirilen, gri ve gri olmayan ısıl ışınım modellerinin koda akuple edilmesi ile gerçekleştirilmiştir. Kimyasal tepkimeleri hesaba katmak için birinci derece (global) tepkime mekanizması kullanılmıştır.

Algoritmanın, takip eden iki durum için: *i)* eş ısıda olmayan kargaşalı boru akışı; *ii)* kargaşasız metan-hava yayılım alevi; öngörüleri, deneysel ölçümler ve sayısal

çözümlerle kıyaslanmış, zamana bağlı akışları öngörme kabiliyeti aynı durumlar için gösterilmiştir. Her iki durumda da müsbet kıyaslamalar elde edilmiştir. Isıl ışınımın ve ışınıma ait özelliklerin gri olmayan bir şekilde değerlendirilmesinin etkileri ikinci test durumu için araştırılmıştır. Isıl ışımanın akuple edilmesinin sıcaklık ve hız alanlarını önemli bir şekilde etkilediği fakat kütle öngörülerindeki etkisinin kısıtlı kaldığı bulunmuştur. Her iki ısıl ışınım modeli ile yapılan hesaplamalar göstermiştir ki, bu çalışmada kullanılan gri olmayan ısıl ışınım modeli, gri olan modelle benzer sonuçları üretmektedir fakat çok daha fazla hesaplama zamanı gerektirmektedir. Geliştirilen algoritma, çeşitli akış senaryolarının zamana bağlı benzetişiminde kullanılabilecek verimli ve çok yönlü bir araç olup zamanla değişen kargaşalı ve tepkime içeren akışların hesaplanması yolundaki ilk adımları oluşturmaktadır.

Anahtar Kelimeler: Tekrarsız algoritmalar, Basınca dayalı metodlar, Operatör bölünmesi, Tepkimeli ısıl ışımalı akışlar, Çizgiler metodu.

To my family and Bahar

ACKNOWLEDGEMENTS

I wish to express my deepest sense of gratitude to my mother, father and brother for their unconditional love, neverending support and sacrifices they have made for me throughout my life. Without them none of this would be possible.

I will remain forever indebted to my supervisor Prof. Dr. Nevin Selçuk for giving me the oppurtinity to be a member of her research group and introducing me to the area of computational fluid dynamics. Her unshakable faith in me and dedication to maintaining high standards in her research have been the constant source of motivation. I also would like to thank my co-supervisor Prof. Dr. İ. Hakkı Tuncer and other members of the examining comittee for their valuable suggestions and comments.

I would like thank the members of my research group, particularly to Tanıl Tarhan for everything he taught me and Işıl Ayrancı, Ertan Karaismail and Nihan Karaismail for their support with the radiation part of the study.

This study was partially supported by Middle East Technical University research funding project BAP No. 2004-07-02-00-125. The support is acknowledged.

Finally, I devote these final words of appreciation to my fiancée Bahar for her understanding, endless patience and encouragement when I needed them the most.

TABLE OF CONTENTS

PLAGIARISM iii
ABSTRACT iv
ÖZ vi
DEDICATION viii
ACKNOWLEDGEMENTS ix
TABLE OF CONTENTS x
LIST OF TABLES xiv
LIST OF FIGURES xv
LIST OF SYMBOLS xix
CHAPTER
1 INTRODUCTION 1
1.1 Preamble 1
1.2 Numerical Approaches for the Solution of Transport Equations . 1
1.3 Motivation 8
1.4 Principle Objectives of the Present Investigation 9
2 PHYSICAL MODELING 11
2.1 Preamble 11
2.1.1 Equation of Continuity 12
2.1.2 Equation of Momentum 12
2.1.3 Equation of Energy 13
2.1.4 Equation of Species 15
2.2 Governing Equations in Cylindrical Coordinates 17
2.3 Equation Summary 21

2.4 Initial and Boundary Conditions 23
2.5 Radiative Transfer Equation 26
2.6 Methane-Air Chemical Reaction Mechanism 27
2.6.1 One-Step Reaction Mechanism 29
2.7 Transport and Thermodynamic Properties 30
3 METHODOLOGY 33
3.1 Preamble 33
3.2 Overview of the Splitting Procedure 33
3.3 Semi-Discrete Approach 34
3.4 Splitting Procedure for Reacting Radiating Flows 35
4 NUMERICAL SOLUTION TECHNIQUE 42
4.1 Preamble 42
4.2 Grid Structure 42
4.3 Computation of Spatial Derivatives 44
4.3.1 Discretization of the Convective Terms 47
4.3.2 Discretization of the Diffusive Terms 48
4.3.3 Discretization of the Pressure Gradient Terms 48
4.4 Time Integration 48
4.5 Numerical Solution of the Pressure Equation 50
4.5.1 Specification of Boundary Conditions for Pressure Equation 52
4.6 Computation of Radiative Source Term 53
4.6.1 Discrete Ordinates Method 53
4.6.2 MOL Solution of DOM for Gray Media 55
4.6.3 MOL Solution of DOM for Non-Gray Media 56
4.6.4 Coupling Strategy 57
5 STRUCTURE AND OPERATION OF THE COMPUTER CODE . . 59
5.1 Preamble 59

5.2 Range of Applicability 59
5.3 Structure of the Code 60
5.4 Mode of Operation 60
5.5 CHEMKIN and TRANSPORT Packages 82
5.6 Pre- and Post-Processing 82
5.7 Programming Language and Compilation of the Code 83
6 RESULTS AND DISCUSSION 84
6.1 Preamble 84
6.2 Test Case 1: Turbulent Gas Flow in a Circular Tube With Strong Wall Heating 85
6.2.1 Grid and Time Step Sensitivity Study 86
6.2.2 Steady State Results 86
6.2.3 Transient Results 95
6.3 Test Case 2: Laminar Methane-Air Diffusion Flame 98
6.3.1 Grid and Time Step Sensitivity Study 98
6.3.2 Numerical Results Without Radiation 100
6.3.3 Comparison of Numerical Results With and Without Radiation 113
6.3.4 Comparison of Numerical Results With Gray and Non-gray Radiation 134
6.3.5 Transient Results 136
7 CONCLUSIONS 140
7.1 Suggestions for Future Work 142
REFERENCES 143
APPENDICES
A ORDINATES AND WEIGHTS FOR $\mathbf{S_N}$ APPROXIMATIONS 151
B INPUT FILES 152
B.1 CHEM.INP 152
B.2 MESHGEN.INI 152

B.3 LOCATIONS.INI 153
B.4 DATA.INI 154
C SOURCE CODES 155
C.1 Program COMBINE 155
C.2 Program EXTRACT 169
D REFERENCE TRANSIENT SOLUTIONS 176
CURRICULUM VITAE 179

LIST OF TABLES

2.1 Initial & Boundary Conditions 24

4.1 Boundary conditions for pressure equation 52

A.1 Discrete ordinates for the S_N approximation for axisymmetric cylindrical geometry. 151

LIST OF FIGURES

1.1 Staggered grid arrangement. 3

2.1 Cylindrical space-angle coordinate system in three dimensions. 27

3.1 Overview of the splitting procedure. 34

3.2 Overview of the splitting procedure for reacting radiating flows. 36

4.1 Representative grid structures for (a) CFD code; (b) Radiation code. i and j correspond to grid indices for r and z directions, respectively. . . . 43

5.1 Organization of the computer code. 61

5.2 Algorithm of the main program. 63

5.3 Algorithm of subroutine INITIAL. 64

5.4 Algorithm of subroutine TIME_INTEGRATION. 65

5.5 Algorithm of subroutine MESH_BOUNDARY. 66

5.6 Algorithm of subroutine CHEMKIN_INITIATION. 66

5.7 Algorithm of subroutine DERV_MOMENTUM. 67

5.8 Algorithm of subroutine DERV_ENERGY. 68

5.9 Algorithm of subroutine DERV_SPECIES. 69

5.10 Algorithm of subroutine DIFFUSIVE_FLUX. 69

5.11 Algorithm of subroutine ENERGY_INTEGRATOR. 70

5.12 Algorithm of subroutine SPECIES_INTEGRATOR. 71

5.13 Algorithm of subroutine MOMENTUM_INTEGRATOR. 72

5.14 Algorithm of subroutine Z_COMPONENT_VELOCITY. 73

5.15 Algorithm of subroutine R_COMPONENT_VELOCITY. 74

5.16 Algorithm of subroutines SOURCE_COMPUTATION, PRESSURE_EQUATION, PRESSURE_GRADIENTS. 75

5.17 Algorithm of subroutine PROPERTY_ESTIMATION. 76

5.18 Algorithm of subroutine TEMPERATURE. 77

5.19 Algorithm of subroutine SPECIES. 78

5.20 Algorithm of subroutine CHEMICAL_REACTION_MECHANISM. . 79
5.21 Algorithm of subroutine DLG4CC. 80
5.22 Algorithm of subroutine DLG2VL2D. 81

6.1 Schematic representation of the test rig and operating conditions. . . . 87
6.2 Radial profiles of mean axial velocity and temperature at z/D = 14.2 computed with different grid resolutions. 88
6.3 Radial profiles of mean axial velocity at three axial locations. 91
6.4 Radial profiles of temperature at three axial locations. 92
6.5 Axial pressure distribution along the centerline. 93
6.6 Contours of axial and radial velocity, temperature and pressure. 94
6.7 Time development of axial velocity field. 96
6.8 Time development of temperature field. 97
6.9 Schematic representation of the co-flowing jet diffusion flame and operating conditions. 99
6.10 Radial profiles of axial velocity, temperature, mole fractions of CO_2 and H_2O at z = 2.4 cm computed with different grid resolutions. 101
6.11 Radial and axial profiles of temperature at three axial locations and along the centerline. 105
6.12 Radial and axial profiles of axial velocity at three axial locations and along the centerline. 106
6.13 Radial and axial profiles of CH_4 mole fractions at three axial locations and along the centerline. 107
6.14 Radial and axial profiles of O_2 mole fractions at three axial locations and along the centerline. 108
6.15 Radial and axial profiles of CO_2 mole fractions at three axial locations and along the centerline. 109
6.16 Radial and axial profiles of H_2O mole fractions at three axial locations and along the centerline. 110
6.17 Radial and axial profiles of N_2 mole fractions at three axial locations and along the centerline. 111
6.18 Radial profiles of radial velocity at two axial locations. 112
6.19 Radial and axial profiles of temperature at three axial locations and along the centerline. 115
6.20 Radial and axial profiles of axial velocity at three axial locations and along the centerline. 116

6.21 Radial and axial profiles CH_4 mole fractions at three axial locations and along the centerline. 117

6.22 Radial and axial profiles O_2 mole fractions at three axial locations and along the centerline. 118

6.23 Radial and axial profiles CO_2 mole fractions at three axial locations and along the centerline. 119

6.24 Radial and axial profiles H_2O mole fractions at three axial locations and along the centerline. 120

6.25 Radial and axial profiles N_2 mole fractions at three axial locations and along the centerline. 121

6.26 Conductive and radiative heat fluxes in *r*-direction at the tip of the flame. 122

6.27 Comparison of temperature contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). . . . 125

6.28 Comparison of axial velocity contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). . . . 126

6.29 Comparison of radial velocity contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). . . . 127

6.30 Comparison of pressure contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model. 128

6.31 Comparison of CH_4 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). 129

6.32 Comparison of O_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). 130

6.33 Comparison of CO_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). 131

6.34 Comparison of H_2O isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). 132

6.35 Comparison of N_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation). 133

6.36 Comparison of divergence of heat flux contours: (a) Present study with gray radiation model; (b) Present study with non-gray radiation model. 135

6.37 Time development of streamline pattern and axial velocity obtained with gray radiation model. 138

6.38 Time development of temperature field obtained with gray radiation model. 139

D.1 Time development of streamline pattern and axial velocity obtained by Uygur *et al.* [2]. 177

D.2 Time development of temperature field obtained by Uygur *et al.* [2]. . . 178

LIST OF SYMBOLS

a	Gray gas weights
A	Perpendicular area to the main flow direction
C_p	Specific heat capacity at constant pressure in molar units
$\hat{C}_p$	Specific heat capacity at constant pressure in mass units
D	Diffusion coefficient
e	Unit vector
$\mathbf{g}$	Gravitational acceleration
H	Enthalpy in molar units
$\hat{H}$	Enthalpy per mass
$\bar{H}$	Partial molar enthalpy
I	Radiative intensity
$\mathbf{j}$	Mass fluxes
m	Discrete direction
$\dot{m}$	Mass flow rate
N	Number of total species
NG	Number of gray gases
NR	Number of grid points in r-direction
NZ	Number of grid points in z-direction
n	Direction normal to the boundary
p	Pressure
$\mathbf{q}$	Heat flux vector
r	Radial distance
R	Universal gas constant; Radius
s	Distance
T	Temperature
t	Time
u	Instantaneous axial velocity
$\hat{U}$	Internal energy per unit mass
$\mathbf{v}$	Velocity vector
$\hat{V}$	Volume per unit mass

v	Instantaneous radial velocity
W	Molecular weight
$\overline{W}$	Mean molecular weight
X	Mole fraction
Y	Mass fraction
z	Axial distance

Greek Letters

$\boldsymbol{\delta}$	Unit vector
Δ	Macroscopic differential operator
θ	Polar angle
κ	Bulk viscosity; Gray absorption coefficient
λ	Thermal conductivity
μ	Dynamic viscosity; Direction cosine
η	Direction cosine
ξ	Direction cosine
γ	Angular differencing coefficient
ℓ	Index for a discrete direction
ε	Emissivity
δ	Stefan-Boltzman constant
∇	Vector differential operator
ρ	Density
$\dot{\omega}$	Reaction rate in mass units
Ω	Direction vector
$\boldsymbol{\tau}$	Viscous momentum flux tensor
ϕ	Dependent variable; Azimuthal angle

Superscripts

n	Index for time-level
o	Standard state
ref	Reference
t	Transpose
T	Thermal
$+$	Forward

$'$	Incident
$*$	First intermediate time level
$**$	Second intermediate time level
$***$	Third intermediate time level

Subscripts

b	Black body
F	Fuel
i, *j*	Index for spatial coordinate of interest and species
in	Inlet
k	The k^{th} species
m	Mixture averaged
O	Oxidizer
r	r-direction
R	Radiation
w	wall
z	z-direction

CHAPTER 1

INTRODUCTION

1.1 Preamble

The primary purpose of this research effort is to develop an algorithm for the numerical simulation of reacting radiating flows. The investigation centers around the development of a non-iterative pressure based scheme which uses the concept of operator-splitting for the solution of governing transport equations. Radiative heat transfer computations are carried out by means of incorporating previously developed gray and non-gray radiation models into the new algorithm.

Below, the reader will find some of the important studies selected from the open literature having relevance to the present investigation and which can be considered as the milestones in the evolution of the widely used numerical approaches. The recent trends in the field will also be demonstrated by the selected studies. Finally, this chapter will be concluded by stating the motivation behind the present study and its objectives.

1.2 Numerical Approaches for the Solution of Transport Equations

The foundations of experimental fluid dynamics were laid in the seventeenth century which then followed by the gradual development of theoretical fluid dynamics in the eighteenth and nineteenth centuries. Until the mid-twentieth century, fluid dynamics was studied and practiced by pure theory on one hand and pure experiment on the other. However, the advent of high speed digital computers combined with the development of accurate numerical algorithms for solving physical problems on

these computers, has revolutionized the discipline of fluid dynamics by introducing a third approach called the computational fluid dynamics (CFD). As the name implies, CFD can generally be defined as solution of governing equations which describe the dynamics of fluids via numerical methods. Yet it should be noted that CFD is not an alternative to experimental methods but it is a complementary approach. However, the fact that it can provide detailed and comprehensive information, that can generally be obtained by experimental methods, in a cost effective manner has led more and more research effort to focus on the utilization of CFD as a tool for both fundamental research and real life problems.

The common numerical techniques exploited by the CFD algorithms for the solution of partial differential equations (PDE) are the classical finite difference method (FDM), finite volume method (FVM) and finite element method (FEM). Historically, FDMs have dominated the CFD community owing to their simplicity in formulations and computations. One of the earliest studies for the solution of incompressible Navier-Stokes equations using FDM was carried out by Harlow and Welch [3]. The novelty brought by the authors was that in the course of the unsteady solution of momentum equations, pressure was calculated from a Poisson type equation which is derived by taking the divergence of the discrete momentum equations. The continuity equation was indirectly satisfied through the solution of pressure equation (pressure correction). Neumann-type boundary conditions necessary for the solution were obtained by using momentum equations at boundaries. Another novelty was the utilization of staggered grid arrangement illustrated in Figure 1.1. As can be seen from the figure, velocity components are distributed around the pressure points in contrast to regular grid system where all dependent variables are stored at the same point. Solution of pressure equation was carried out on the pressure node whereas the momentum equation corresponding to each velocity component was solved at the respective velocity point. The purpose of such an organization was to remove the spurious oscillations in pressure and consequently in velocities (known as the checkerboard-type oscillations or odd-even decoupling) resulting from the utilization of centered schemes on regular grids. The method was successfully applied to time-

dependent solution of various flow problems.

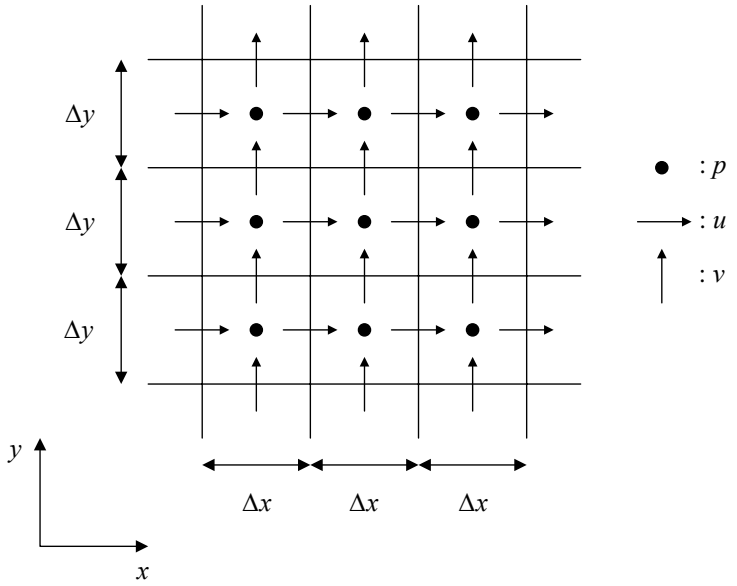


Figure 1.1: Staggered grid arrangement.

Another pioneering study based on staggered grids and Poisson type equation for pressure was performed by Patankar and Spalding [4]. The method was called SIMPLE which stands for Semi-Implicit Method for Pressure-Linked Equations and was originally proposed for incompressible flows. The SIMPLE scheme is formulated in terms of a pressure-correction variable, the difference between predicted and corrected pressures. An iterative loop is employed between the discrete pressure Poisson equation and the momentum equations. Convergence is achieved when the pressure correction is globally near zero.

Over the years, SIMPLE and its variants SIMPLER [5], SIMPLEC [6], SIMPLEX [7] were used extensively by numerous researchers for the predcition of steady incompressible flows. The compressible version of the scheme developed by Van Doormaal *et al.* [8] however, was not as attractive as its incompressible predecessor due to the presence of an outer loop to account for the energy effects on top of the

inner loop.

Solution methods for fluid flow equations can be broadly categorized as density based and pressure based methods [9]. As the name implies the former treats density as the main dependent variable and pressure is calculated via an equation of state [10] while the roles of pressure and density are reversed in the latter. Owing to the fact that the link between density and pressure is weak in incompressible or weakly compressible flows, application of density based methods is limited to compressible flows. Pressure based methods on the other hand, are technically suitable for all flow regimes ranging from incompressible to compressible. Due to the fact that pressure (itself or in the form of pressure-correction) is designated as one of the primary variables in the abovementioned studies, they all fall into category of pressure-based methods [9].

Among the pressure based methods, PISO [11, 12] which stands for pressure-implicit with splitting operator has proved to be an accurate and efficient scheme for transient solutions due to its non-iterative nature. PISO exploits the concept of factorization (splitting of operators) by extending it to the coupling between pressure and velocity whereby operations involving different variables are split into a series of predictor-corrector stages (one predictor and two correctors). The scheme commences with an implicit momentum predictor stage, solution of which does not necessarily satisfy the equation of continuity. In the succeeding explicit corrector stages, a Poisson type equation for pressure is solved on staggered grid topology which ensures that the momentum equations solved at the corrector stages satisfy the equation of continuity. The predictive performance of the method was benchmarked with various incompressible/compressible flow problems [12, 13].

The common feature of all the studies mentioned so far was that they are all based on staggered grids. However, the drawbacks of formulations which utilize staggered grids, especially in curvilinear coordinates, are that they are inherently complicated and require additional effort in bookkeeping and boundary condition implementation. In a study by Rhei and Chow [14], the need for staggered grids was eluded by the use

of collocated grid system. Oscillation-free pressure field was obtained at the expense of defining new variables called the flux velocities at the cell interfaces, calculation of which necessitates interpolation and hence additional cost. It was later shown by Abdallah [15,16] and co-workers [17,18] that regular grids can be used in the solution of incompressible Navier-Stokes equations as well, provided that the compatibility condition which relates the source of the pressure Poisson equation to the Neumann boundary conditions is satisfied.

Regardless of the numerical method used, the objective of the developed algorithms is to be able to handle real life problems. Today, numerical simulation of reacting flows is one of most appealing research topics due to their presence in practical systems. Hence a significant amount of research effort has been devoted to the subject in the open literature [19–27].

In one of these research studies, Issa *et al.* [19] extended the non-iterative PISO scheme [11, 12] to the transient simulation of reacting flows by incorporating the species equations into the predictor-corrector sequence. Although the predictive accuracy was not demonstrated on a realistic problem, it was reported that algorithm reproduces the results obtained with a comparable iterative secheme.

In a similar study by Najm *et al.* [20], a semi-implicit projection scheme based on predictor-corrector approach was developed for the simulation of unsteady combustion in two dimensions. The scheme relies on zero-Mach number formulation of the compressible conservation equations with detailed chemistry (GRImech 1.2 for methane-air combustion involving 32 species and 177 elementary reactions). The study was exclusively focused on overcoming the temporal stiffness due to detailed reaction mechanims and this was achieved by utilizing a stiff integrator for the solution of species equations. For the methane-air premixed flame considered in the study, it was shown that the stiff scheme employed enables the selection of larger time steps and thus leads to substantial improvement in the performance of the computations.

Despite numerous studies carried out on different aspects of reacting flows, numerical

simulation of turbulent reacting flows still remains to be scientific challange due to the difficulty of resolving the very disparate time scales of the controlling physical and chemical processes [28]. The conventional routes for the simulation of turbulent reacting flows are Reynolds averaged Navier-Stokes simulation (RANS), large eddy simulation (LES) and direct numerical simulation (DNS). RANS simulations are carried out by solving averaged transport equations closed with turbulence models. The main drawback of RANS is that it relies on the turbulence closures which are not universal. The DNS approach consists of solving exactly all the spatial and time-scales embedded in the representative flow equations, without any model for turbulence and hence it is the most accurate and straightforward technique. LES can be seen as an intermediate between DNS and RANS. In LES, the largest structures of the flow field are explicitly computed as in the case of DNS whereas the effects of small-scale structures are modeled.

It should be noted that accurate modeling of reacting flows relies not only on the simulation technique but also on the models needed to describe chemical reactions, transport and thermodynamics properties, radiative transfer, soot formation etc. Many different models, with different levels of accuracy and complexity are available in the literature. Undoubtedly, DNS of turbulent reacting flows with detailed models is expected to give the most comprehensive information on the flow field. However, the fact that this is still beyond the capabilities of current computational resources enforces the researchers to make a compromise between the level of detail of reaction mechanism and incorporation of models for radiation transport and soot formation.

Such a compromise was clearly demonstrated in a study by Bedat *et al.* [24] on the DNS of turbulent methane-air diffusion flames. Single to multiple step chemistry including NO_x formation were employed in the computations. The predictions were benchmarked against the results obtained by detailed chemistry (GRI 2.11). It was found that at least a four-step mechanism was required in order to make favorable comparisons.

In another study, the effect of radiation and soot models on transient simulation

of ethylene-air turbulent jet diffusion flames whilst using a single step reaction mechanism was investigated by Kaplan *et al.* [28]. Discrete ordinates method (DOM) with S_4 approximation along with gray gas assumption was utilized for the solution of radiative transfer equation (RTE). The major goal of the study was to assess the importance of radiation transport on the dynamics of strongly radiating luminous flames. The simulations showed that radiative heat losses reduce the flame temperature which decreases the chemical heat release rate causing the flame to shrink considerably and hence change the overall temperature, species concentration and soot volume fraction distributions in the flame. It was further suggested that radiative transport dominates over transfer by conduction and convection in the heavily sooting region.

Coelho *et al.* [29] studied the effect spectral radiative effects and turbulence/radiation interaction in a non-luminous methane/air turbulent jet diffusion flame on which experimental data is available. For radiative transport calculations, the medium was either treated as gray using the Planck mean absorption coefficient or non-gray using (SLW) model together with the optically thin approximation. The comparisons between the simulations and the experimental results showed that the radiative heat loss is significantly overestimated in both gray and non-gray cases. However, the results obtained for the non-gray case were found to be closer to the experimental data.

While the ultimate goal in combustion research is the DNS of turbulent reacting flows, simulation of laminar reacting flows as an intermediate step towards this goal is still the objective of present studies. Recently, Liu *et al.* [30] showed the effects of radiation models on the modeling of laminar co-flowing methane-air diffusion flame studied by Smooke *et al.* [31]. Detailed steady-state numerical calculations were performed using complex thermal and transport properties. GRI-Mech 3.0 without reactions and species related to NO_X formation resulting in a mechanism consisting of 219 reactions and 36 species was utilized for the chemical reaction mechanism. Soot kinetics was also incorporated. Nongray radiative heat transfer by CO_2, H_2O, CO, and soot was calculated using the discrete-ordinates method (DOM)

coupled with several statistical-narrow-band-based correlated-*k* (SNBCK) models. The calculated temperature and soot volume fraction distributions were compared with the experimental data [32]. Excellent agreement were observed in the centerline region, however, a relatively large discrepancy was found near the flame edge. The calculated radial temperature and soot volume fractions at several heights were in much better agreement with the experimental data than the predictions of Smooke *et al.* [31]. The effect of radiation absorption in methane-air flames was found to be relatively unimportant as compared to more heavily sooting flames and it was also found that neglecting radiation leads to a cooler flame with a maximum error of about 17 K in contrast to the error of 122 K reported by Smooke *et al.* [31]. Computational efficiency of the radiation models were also compared. DOM/optimized 9-band using four-quadrature was found to be very accurate and efficient and recommended for the calculation of radiative heat transfer in sooting flames.

1.3 Motivation

Considering the emphasis on the prediction of transient turbulent reacting flows, a novel CFD code based on Method of Lines (MOL) was developed in Middle East Technical University Chemical Engineering Department, for the unsteady simulation of 2D incompressible, separated, internal, non-isothermal flows in regular and complex geometries [33]. The code uses MOL, which is an efficient semi-discrete approach for the solution of time-dependent partial differential equations (PDEs), in conjunction with: *i)* a higher-order spatial discretization scheme which chooses biased-upwind or biased-downwind schemes in a zone of dependence manner; *ii)* a parabolic algorithm for the computation of axial pressure gradient which does not require the solution of an elliptic equation for pressure; *iii)* an elliptic grid generator using body-fitted coordinate system for application to complex geometries. The validity and the predictive ability of the code were tested by applying it to the simulation of laminar/turbulent, isothermal/non-isothermal incompressible flows and comparing its predictions with either measured data or numerical simulations available in the literature [34–43]. In successive studies by the same group [1, 2], the code was further developed by incorporation of the solution of species equations

using finite rate chemistry model together with a Total Variation Diminishing (TVD) flux limiter based discretization scheme for the computation of convective derivatives [1]and a radiation submodel to account for radiative heat transfer [2], for the simulation of transient reacting radiating flows. The predictive performance of the code was tested by applying it to the simulation of a confined laminar methane/air diffusion flame and comparing its predictions with numerical and experimental data available in the literature [44–46]. The velocity, temperature and major species concentrations obtained with and without radiation model were found to be in reasonably good agreement with numerical results and measurements [2].

Although providing a useful basis for the simulation of reacting flows, the code used in [1, 2] is limited to incompressible flows and its extension to three dimensions is not possible due to the parabolic pressure scheme [34] embedded in the flow solver. The parabolic scheme only allows z-component of momentum equation to be solved and r-component of velocity is calculated by direct utilization of continuity equation by dropping the time derivative of density, an approach which can only be valid if the flow field is treated as incompressible. Considering the vast density changes due to temperature and concentration variations, more accurate representation of typical flows necessitates pressure based schemes which can take into account the compressibility effects.

1.4 Principle Objectives of the Present Investigation

The present study focuses on the development of a non-iterative pressure based algorithm for the transient simulation of reacting radiating flows as an initial step towards the time-dependent computation of turbulent combustion. Similar to PISO approach developed for reacting flows [19], the new algorithm is based on a sequence of predictor and corrector stages for momentum, energy and species equations. A pressure equation which replaces the role of equation of continuity is solved at momentum corrector stages by means of a multigrid solver, for the determination of pressure field. The differences between the proposed algorithm and PISO lie in the time-integration method and the grid topology utilized. In the present approach,

the conservation equations are cast in their semi-discrete form using finite difference approximations on non-staggered grid topology which results in a system of ODEs. The resulting ODEs are integrated in time using higher-order, implicit algorithms embedded in the sophisticated ODE solvers. By this way, the present algorithm not only offers implicit hence stable time-integration at all splitting phases without extra complexity in the formulation, but also the flexibility and modularity to incorporate any desired package with ease. Moreover, with the utilization of non-staggered grid topology, easier book-keeping is maintained throughout the algorithm as opposed to staggered one and the feasibility of its application to complex geometries is greatly enhanced.

In the course of the development of the algorithm, the following stages have been followed:

- Development of a non-iterative pressure based algorithm for the solution of momentum and energy equations,
- Benchmarking the predictions of the algorithm for non-isothermal flows against experimental data [47] and numerical solutions [43, 48] available in the literature,
- Incorporation of previously developed gray [49] and non-gray radiation [50] models into the algorithm for the computation of radiative heat transfer,
- Completion of the algorithm by the incorporation of the solution of species equations,
- Validation of the predictions of the complete algorithm against experimental data [45, 46] and numerical solutions [1, 2] available in the literature.

CHAPTER 2

PHYSICAL MODELING

2.1 Preamble

The physical system to be modelled is a confined, vertical, two-dimensional, laminar methane-air diffusion flame at atmospheric pressure. In this chapter, governing equations of fluid dynamics, simplifying assumptions employed, transport and thermodynamic models utilized, and initial and boundary conditions required for a complete representation of the multi-component reacting system under consideration will be presented.

The fundamental equations of fluid dynamics for chemically reacting flows in multi-component systems are based on the following conservation laws:

1. Conservation of mass,
2. Conservation of momentum,
3. Conservation of energy.

Application of conservation of mass to a volume element within the flowing mixture results in the continuity equation. Conservation of momentum is represented by momentum equation which is the interpretation of Newton's second law of motion for fluid flow. The law of conservation of energy is identical to first law of thermodynamics and the resulting fluid dynamics equation is called the energy equation. Application of conservation of mass to each species within the flowing mixture yields the species equation.

In the following sections, these equations are described in detail.

2.1.1 Equation of Continuity

Equation of continuity describes the time rate of the mixture density at a fixed point in space:

$$\underbrace{\frac{\partial}{\partial t}\rho}_{\substack{\text{rate of increase}\\ \text{of mass per unit}\\ \text{volume}}} = \underbrace{-(\nabla \cdot \rho \mathbf{v})}_{\substack{\text{net rate of mass}\\ \text{addition per unit}\\ \text{volume}}}. \tag{2.1}$$

The term on the right-hand-side of Equation (2.1) is called the divergence of the mass flux vector ($\rho\mathbf{v}$) and accounts for the net rate of mass efflux per unit volume.

2.1.2 Equation of Momentum

The equation of momentum can be written in vector-tensor notation as

$$\underbrace{\frac{\partial}{\partial t}\rho\mathbf{v}}_{\substack{\text{rate of increase}\\ \text{of momentum}\\ \text{per unit volume}}} = \underbrace{-[\nabla \cdot \rho\mathbf{v}\mathbf{v}]}_{\substack{\text{rate of momentum}\\ \text{gain by convection}\\ \text{per unit volume}}} \underbrace{-\nabla p}_{\substack{\text{pressure force}\\ \text{on mixture per}\\ \text{unit volume}}} \underbrace{-[\nabla \cdot \boldsymbol{\tau}]}_{\substack{\text{rate of momentum}\\ \text{gain by molecular}\\ \text{transport per unit}\\ \text{volume}}} \underbrace{+\rho\mathbf{g}}_{\substack{\text{external force on}\\ \text{mixture per unit}\\ \text{volume}}}. \tag{2.2}$$

In this equation, ∇p is a vector called the pressure gradient, the vector $[\nabla \cdot \boldsymbol{\tau}]$ is the divergence of the viscous stress tensor $\boldsymbol{\tau}$ and the vector $[\nabla \cdot \rho\mathbf{v}\mathbf{v}]$ is the divergence of the dyadic product $\rho\mathbf{v}\mathbf{v}$. The viscous stress tensor $\boldsymbol{\tau}$ is related to velocity gradients through to the following expression based on the kinetic theory

$$\boldsymbol{\tau} = -\mu(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathbf{t}}) + (\tfrac{2}{3}\mu - \kappa)(\nabla \cdot \mathbf{v})\boldsymbol{\delta} \tag{2.3}$$

in which μ is the dynamic viscosity, κ is the bulk viscosity, $\boldsymbol{\delta}$ is the unit tensor with components δ_{ij}, $\nabla\mathbf{v}$ is the velocity gradient tensor with components $(\partial/\partial x_i)\upsilon_j$, $(\nabla\mathbf{v})^{\mathrm{t}}$ is the transpose of the velocity gradient tensor with components $(\partial/\partial x_j)\upsilon_i$, and $(\nabla \cdot \mathbf{v})$ is the divergence of the velocity vector. The bulk viscosity term κ is often assumed to be zero for gases and hence Equation (2.3) becomes

$$\boldsymbol{\tau} = -\mu(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{t}}) + \tfrac{2}{3}\mu(\nabla \cdot \mathbf{v})\boldsymbol{\delta}. \tag{2.4}$$

2.1.3 Equation of Energy

The general form of the energy equation written in vector-tensor notation is as follows

$$\underset{\substack{\text{rate of increase of}\\ \text{energy per unit}\\ \text{volume}}}{\frac{\partial}{\partial t}\rho(\hat{U}+\tfrac{1}{2}v^2)} = \underset{\substack{\text{rate of energy addition per}\\ \text{unit volume by convective}\\ \text{transport}}}{-[\nabla\cdot\rho\mathbf{v}(\hat{U}+\tfrac{1}{2}v^2)]} \underset{\substack{\text{rate of energy}\\ \text{addition per unit}\\ \text{volume by heat}\\ \text{conduction}}}{-(\nabla\cdot\mathbf{q})} \underset{\substack{\text{rate of work done on the}\\ \text{mixture per unit volume}\\ \text{by pressure forces}}}{-[\nabla\cdot p\mathbf{v}]} \underset{\substack{\text{rate of work done on the}\\ \text{mixture per unit volume}\\ \text{by viscous forces}}}{-(\nabla\cdot[\tau\cdot\mathbf{v}])} \underset{\substack{\text{rate of work done on the}\\ \text{mixture per unit volume by}\\ \text{external forces}}}{+\rho(\mathbf{v}\cdot\mathbf{g}).} \tag{2.5}$$

It should be noted that this equation does not contain a source term to describe the thermal energy released by homogeneous chemical reactions as it is included implicitly in function $\hat{U}$. A more convenient form of the energy equation is the one which is expressed in terms of temperature. In order to obtain such an equation, first the *equation of mechanical energy*

$$\underset{\substack{\text{rate of increase}\\ \text{of kinetic energy}\\ \text{per unit volume}}}{\frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2)} = \underset{\substack{\text{rate of addition of}\\ \text{kinetic energy by}\\ \text{convection per unit}\\ \text{volume}}}{-[\nabla\cdot\rho\tfrac{1}{2}v^2\mathbf{v}]} \underset{\substack{\text{rate of work done}\\ \text{by pressure of}\\ \text{surroundings on}\\ \text{the mixture}}}{-(\nabla\cdot p\mathbf{v})} \underset{\substack{\text{rate of reversible}\\ \text{conversion of}\\ \text{kinetic energy into}\\ \text{internal energy}}}{-p(-\nabla\cdot\mathbf{v})} \underset{\substack{\text{rate of work done}\\ \text{by viscous forces}\\ \text{on the mixture}}}{-(\nabla\cdot[\tau\cdot\mathbf{v}])} \underset{\substack{\text{rate of}\\ \text{irreversible}\\ \text{conversion from}\\ \text{kinetic to internal}\\ \text{energy}}}{-(-\tau:\nabla\mathbf{v})} \underset{\substack{\text{rate of work done}\\ \text{by external force}\\ \text{on the mixture}}}{+\rho(\mathbf{v}\cdot\mathbf{g})} \tag{2.6}$$

is subtracted from Equation (2.5) which yields the *equation of thermal energy*

$$\underset{\substack{\text{rate of}\\ \text{increase of}\\ \text{internal energy}\\ \text{per unit}\\ \text{volume}}}{\frac{\partial}{\partial t}\rho\hat{U}} = \underset{\substack{\text{net rate of}\\ \text{addition of}\\ \text{internal energy}\\ \text{by convective}\\ \text{transport per unit}\\ \text{volume}}}{-(\nabla\cdot\rho\hat{U}\mathbf{v})} \underset{\substack{\text{rate of energy}\\ \text{addition by}\\ \text{heat}\\ \text{conduction per}\\ \text{unit volume}}}{-(\nabla\cdot\mathbf{q})} \underset{\substack{\text{reversible rate of}\\ \text{internal energy}\\ \text{increase per unit}\\ \text{volume by}\\ \text{compression}}}{-p(\nabla\cdot\mathbf{v}]} \underset{\substack{\text{irreversible rate of}\\ \text{internal energy}\\ \text{increase per unit}\\ \text{volume by viscous}\\ \text{dissipation}}}{-(\tau:\nabla\mathbf{v}).} \tag{2.7}$$

Expressing internal energy in terms of enthalpy using the following relation

$$\hat{U}=\hat{H}+p\hat{V}=\hat{H}-(p/\rho) \tag{2.8}$$

one obtains

$$\frac{\partial}{\partial t}\rho\hat{H} = -(\nabla\cdot\rho\hat{H}\mathbf{v}) - (\nabla\cdot\mathbf{q}) - (\tau:\nabla\mathbf{v}) + \mathbf{v}\cdot\nabla p + \frac{\partial p}{\partial t}. \tag{2.9}$$

Utilizing the following standard equilibrium thermodynamics formula which relates enthalpy to temperature

$$d\hat{H} = \hat{C}_p dT + \left[\hat{V} - T\left(\frac{\partial\hat{V}}{\partial T}\right)_p\right] dp \tag{2.10}$$

and neglecting pressure terms for the laminar flow under consideration, Equation (2.9) becomes

$$\hat{C}_p\frac{\partial}{\partial t}\rho T = -\hat{C}_p(\nabla\cdot\rho\mathbf{v}T) - (\nabla\cdot\mathbf{q}) - (\tau:\nabla\mathbf{v}) + \sum_{k=1}^{N}\hat{H}_k(\nabla\cdot\mathbf{j}_k) - \sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k. \tag{2.11}$$

The last two terms appearing in Equation (2.11) are the source terms for the volumetric production of internal energy due to diffusion and the volumetric production of heat due to chemical reactions, respectively. These terms included implicitly in function $\hat{U}$ are introduced when energy equation is written in terms of temperature.

For multi-component mixtures, the heat flux vector $\mathbf{q}$ given below consists of four terms.

$$\mathbf{q} = -\lambda\nabla T + \sum_{k=1}^{N}\frac{\bar{H}_k}{W_k}\mathbf{j}_k + \sum_{k=1}^{N}\frac{RT}{W_k X_k}D_k^T\mathbf{d}_k + \mathbf{q_R} \tag{2.12}$$

The first term is the heat transport by conduction based on Fourier's Law where λ is the thermal conductivity of the mixture. The second term describes the heat transport by each diffusing species. The third term also known as Dufour term represents the heat transfer produced due to the concentration gradient. The final term accounts for the radiative energy transport.

Among the terms which constitute the heat flux vector, heat conduction, heat diffusion and radiative energy transport are the important ones whereas the Dufour term is usually small when compared to the former three and can ben neglected. Hence Equation (2.12) becomes

$$\mathbf{q} = -\lambda\nabla T + \sum_{k=1}^{N}\frac{\bar{H}_k}{W_k}\mathbf{j}_k + \mathbf{q_R}. \tag{2.13}$$

For ideal gas mixtures, this expression can be further simplified by replacing the partial molar enthalpies, $\bar{H}_k$, by the molar enthalpies, $\hat{H}_k$, yielding

$$\mathbf{q} = -\lambda \nabla T + \sum_{k=1}^{N} \hat{H}_k \, \mathbf{j}_k + \mathbf{q_R}. \tag{2.14}$$

Substituting Equation (2.14) into Equation (2.11) and neglecting the viscous dissipation term lead to

$$\begin{aligned} \hat{C}_p \frac{\partial}{\partial t} \rho T = & - \hat{C}_p (\nabla \cdot \rho \mathbf{v} T) + \nabla \cdot (\lambda \nabla T) \\ & - \sum_{k=1}^{N} \hat{H}_k \, \mathbf{j}_k + \sum_{k=1}^{N} \hat{H}_k (\nabla \cdot \mathbf{j}_k) - \sum_{k=1}^{N} \hat{H}_k \dot{\omega}_k - \nabla \cdot \mathbf{q_R}. \end{aligned} \tag{2.15}$$

Combining the third and fourth terms appearing on the right hand side of Equation (2.15) to yield a single heat source term due to diffusive flux, i.e.,

$$- \sum_{k=1}^{N} \hat{H}_k \, \mathbf{j}_k + \sum_{k=1}^{N} \hat{H}_k (\nabla \cdot \mathbf{j}_k) = - \sum_{k=1}^{N} \mathbf{j}_k \cdot \hat{C}_{p,k} \nabla T \tag{2.16}$$

results in the final form of energy equation

$$\begin{aligned} \hat{C}_p \frac{\partial}{\partial t} \rho T = & - \hat{C}_p (\nabla \cdot \rho \mathbf{v} T) + \nabla \cdot (\lambda \nabla T) \\ & - \sum_{k=1}^{N} \mathbf{j}_k \cdot \hat{C}_{p,k} \nabla T - \sum_{k=1}^{N} \hat{H}_k \dot{\omega}_k - \nabla \cdot \mathbf{q_R}. \end{aligned} \tag{2.17}$$

2.1.4 Equation of Species

The equation of species in vector-tensor notation can be written as follows

$$\underset{\substack{\text{rate of increase of mass of species } k \text{ per unit volume}}}{\frac{\partial}{\partial t} \rho Y_k} = \underset{\substack{\text{net rate of addition of mass of species } k \text{ per unit volume by convection}}}{-(\nabla \cdot \rho \mathbf{v} Y_k)} \quad \underset{\substack{\text{net rate of addition of mass of species } k \text{ per unit volume by diffusion}}}{-(\nabla \cdot \mathbf{j}_k)} \quad \underset{\substack{\text{rate of production of mass of species } k \text{ per unit volume by reaction}}}{+\dot{\omega}_k.} \qquad k = 1, ..., N \tag{2.18}$$

The mass flux vector $\mathbf{j}_k$ in a multi-component system can be expressed by

$$\mathbf{j}_k = \frac{\rho_k}{X_k \overline{W}} \sum_{j \neq k}^{N} W_j D_{kj} \mathbf{d}_j - D_k^T \frac{\nabla T}{T} \tag{2.19}$$

where D_{kj} is the multi-component diffusion coefficient, D_k^T is the thermal diffusion coefficient of species k. Diffusional driving forces $\mathbf{d}_j$ in the first term of Equation (**??**) take into account the three contributions associated with mechanical forces:

$$\mathbf{d}_j = \nabla X_j + (X_j - Y_j) \frac{\nabla p}{p} + \frac{\rho}{p} \sum_{i=1}^{N} Y_i \, Y_j (\mathbf{g}_i - \mathbf{g}_j). \tag{2.20}$$

The terms on the right hand side of this equation are concentration diffusion term, pressure diffusion term and forced diffusion term, respectively. The pressure diffusion term indicates that there may be a net movement of the k^{th} species in a mixture if there is a pressure gradient imposed on the system. This term is usually negligible compared to other terms. The forced diffusion term is of primary importance in ionic systems. If gravity is the only external force then all the $\mathbf{g}_i$ are the same and this term is identically zero. The second term in Equation (2.19) represents the Soret effect accounting for the diffusion of mass due to temperature gradients. This effect tends to drive light molecules towards hot regions and heavy molecules towards cold region of the mixture. It is neglected as it is relatively expensive in terms of computing times. With these assumptions, Equation (2.19) simply becomes

$$\mathbf{j}_k = \frac{\rho_k}{X_k \overline{W}} \sum_{j \neq k}^{N} W_j D_{kj} \nabla X_j \tag{2.21}$$

Multi-component diffusion coefficients D_{kj} are non-linear functions of the local composition, temperature and pressure of the mixture and hence their evaluation is very costly in terms of CPU time. Hence, Equation (2.21) is related to the species gradients by a Fickian formula as

$$\mathbf{j}_k = -\rho_k \frac{D_{km}}{X_k} \nabla X_k \tag{2.22}$$

Here, D_{km} is the mixture–averaged diffusion coefficients between species k and the remaining mixture. The species mole fractions X_k are related to mass fraction Y_k by

$$X_k = Y_k \frac{\overline{W}}{W_k} \tag{2.23}$$

Using this relation Equation (2.22) can be written in terms of Y_k as

$$\mathbf{j}_k = -\rho_k \frac{D_{km}}{Y_k} \nabla Y_k \tag{2.24}$$

or

$$\mathbf{j}_k = -\rho D_{km} \nabla Y_k \tag{2.25}$$

The mixture–averaged diffusion formula does not necessarily satisfy the condition that the sum of the diffusive fluxes is zero, i.e, the condition,

$$\sum_{k=1}^{N} \mathbf{j}_k = 0 \tag{2.26}$$

Therefore, a correction is necessary to ensure mass conservation. For this purpose, rather than solving the species equation for the excess species, its mass fraction is computed simply by

$$Y_N = 1 - \sum_{k=1}^{N-1} Y_k \tag{2.27}$$

The diffusive flux of excess species is computed by the following formula to ensure the mass conservation constraint (Equation (2.26)):

$$\mathbf{j}_N = -\sum_{k=1}^{N-1} \mathbf{j}_k \tag{2.28}$$

2.2 Governing Equations in Cylindrical Coordinates

For two-dimensional, axi-symmetric, laminar reacting radiating flows, governing equations in primitive variables form can be written in cylindrical coordinates as follows;

continuity;

$$\frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(\rho r v) + \frac{\partial}{\partial z}(\rho u) = 0, \tag{2.29}$$

r-momentum;

$$\begin{aligned} \frac{\partial}{\partial t}(\rho v) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho vv) + \frac{\partial}{\partial z}(\rho uv) &= -\frac{\partial p}{\partial r} \\ &- \left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rr}) + \frac{\partial \tau_{zr}}{\partial z} - \frac{1}{r}\tau_{\theta\theta}\right) + \rho g_r, \end{aligned} \tag{2.30}$$

z-momentum;

$$\begin{aligned} \frac{\partial}{\partial t}(\rho u) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho uv) + \frac{\partial}{\partial z}(\rho uu) &= -\frac{\partial p}{\partial z} \\ &- \left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rz}) + \frac{\partial \tau_{zz}}{\partial z}\right) + \rho g_z, \end{aligned} \tag{2.31}$$

energy;

$$\begin{aligned} \hat{C}_p\left(\frac{\partial}{\partial t}(\rho T) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho vT) + \frac{\partial}{\partial z}(\rho uT)\right) &= \left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda\frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(\lambda\frac{\partial T}{\partial z}\right)\right] \\ &- \sum_{k=1}^{N} \hat{C}_{p,k}\left(j_{k,r}\frac{\partial T}{\partial r} + j_{k,z}\frac{\partial T}{\partial z}\right) \\ &- \sum_{k=1}^{N} \hat{H}_k \dot{\omega}_k \\ &- \nabla \cdot \mathbf{q_R}, \end{aligned} \tag{2.32}$$

species;

$$\frac{\partial}{\partial t}(\rho Y_k)+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v Y_k)+\frac{\partial}{\partial z}(\rho u Y_k)=-\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r j_{k,r}\right)+\frac{\partial j_{k,z}}{\partial z}\right)+\dot{\omega}_k. \quad (2.33)$$

The stress tensor components appearing in Equations (2.30) and (2.31) are expressed for Newtonian fluids considered in the present study are;

$$\tau_{rr}=-\mu\left(2\frac{\partial v}{\partial r}\right)+\frac{2}{3}\mu(\nabla\cdot\mathbf{v}), \quad (2.34)$$

$$\tau_{zz}=-\mu\left(2\frac{\partial u}{\partial z}\right)+\frac{2}{3}\mu(\nabla\cdot\mathbf{v}), \quad (2.35)$$

$$\tau_{\theta\theta}=-\mu\left(2\frac{v}{r}\right)+\frac{2}{3}\mu(\nabla\cdot\mathbf{v}), \quad (2.36)$$

$$\tau_{rz}=\tau_{zr}=-\mu\left(\frac{\partial u}{\partial r}+\frac{\partial v}{\partial z}\right), \quad (2.37)$$

where

$$\nabla\cdot\mathbf{v}=\frac{1}{r}\frac{\partial}{\partial r}(rv)+\frac{\partial u}{\partial z}. \quad (2.38)$$

Expanding the derivatives appearing on the left hand side of Equations (2.30)-(2.33) gives

r-momentum;

$$\rho\frac{\partial v}{\partial t}+v\frac{\partial \rho}{\partial t}+\frac{v}{r}\frac{\partial}{\partial r}(r\rho v)+\rho v\frac{\partial v}{\partial r}+v\frac{\partial}{\partial z}(\rho u)+\rho u\frac{\partial v}{\partial z}= -\frac{\partial p}{\partial r}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rr})+\frac{\partial \tau_{zr}}{\partial z}-\frac{1}{r}\tau_{\theta\theta}\right)+\rho g_r, \quad (2.39)$$

z-momentum;

$$\rho\frac{\partial u}{\partial t}+u\frac{\partial \rho}{\partial t}+\frac{u}{r}\frac{\partial}{\partial r}(r\rho v)+\rho v\frac{\partial u}{\partial r}+u\frac{\partial}{\partial z}(\rho u)+\rho u\frac{\partial u}{\partial z}= -\frac{\partial p}{\partial z}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rz})+\frac{\partial \tau_{zz}}{\partial z}\right)+\rho g_z, \quad (2.40)$$

energy;

$$\begin{aligned}\rho\hat{C}_p\frac{\partial T}{\partial t} &+ \hat{C}_p\left(T\frac{\partial \rho}{\partial t}+\frac{T}{r}\frac{\partial}{\partial r}(r\rho v)+\rho v\frac{\partial T}{\partial r}+T\frac{\partial}{\partial z}(\rho u)+\rho u\frac{\partial T}{\partial z}\right)= \\ &\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda\frac{\partial T}{\partial r}\right)+\frac{\partial}{\partial z}\left(\lambda\frac{\partial T}{\partial z}\right)\right] \\ &- \sum_{k=1}^{N}\hat{C}_{p,k}\left(j_{k,r}\frac{\partial T}{\partial r}+j_{k,z}\frac{\partial T}{\partial z}\right) \\ &- \sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k \\ &- \nabla\cdot\mathbf{q_R},\end{aligned} \quad (2.41)$$

species;

$$\rho\frac{\partial Y_k}{\partial t}+Y_k\frac{\partial \rho}{\partial t}+\frac{Y_k}{r}\frac{\partial}{\partial r}(r\rho v)+\rho v\frac{\partial Y_k}{\partial r}+Y_k\frac{\partial}{\partial z}(\rho u)+\rho u\frac{\partial Y_k}{\partial z}= -\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r j_{k,r}\right)+\frac{\partial j_{k,z}}{\partial z}\right)+\dot{\omega}_k. \tag{2.42}$$

Rearranging the terms in Equations (2.39)-(2.42) yields

r-momentum;

$$\rho\left(\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial r}+u\frac{\partial v}{\partial z}\right)+v\left(\frac{\partial \rho}{\partial t}+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v)+\frac{\partial}{\partial z}(\rho u)\right)= -\frac{\partial p}{\partial r}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rr})+\frac{\partial \tau_{zr}}{\partial z}-\frac{1}{r}\tau_{\theta\theta}\right)+\rho g_r, \tag{2.43}$$

z-momentum;

$$\rho\left(\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial r}+u\frac{\partial u}{\partial z}\right)+u\left(\frac{\partial \rho}{\partial t}+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v)+\frac{\partial}{\partial z}(\rho u)\right)= -\frac{\partial p}{\partial z}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rz})+\frac{\partial \tau_{zz}}{\partial z}\right)+\rho g_z, \tag{2.44}$$

energy;

$$\begin{aligned}\rho\hat{C}_p\left(\frac{\partial T}{\partial t}+v\frac{\partial T}{\partial r}+u\frac{\partial T}{\partial z}\right) &+ \hat{C}_pT\left(\frac{\partial \rho}{\partial t}+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v)+\frac{\partial}{\partial z}(\rho u)\right)= \\ &\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda\frac{\partial T}{\partial r}\right)+\frac{\partial}{\partial z}\left(\lambda\frac{\partial T}{\partial z}\right)\right] \\ &- \sum_{k=1}^{N}\hat{C}_{p,k}\left(j_{k,r}\frac{\partial T}{\partial r}+j_{k,z}\frac{\partial T}{\partial z}\right) \\ &- \sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k \\ &- \nabla\cdot\mathbf{q_R},\end{aligned} \tag{2.45}$$

species;

$$\rho\left(\frac{\partial Y_k}{\partial t}+v\frac{\partial Y_k}{\partial r}+u\frac{\partial Y_k}{\partial z}\right)+Y_k\left(\frac{\partial \rho}{\partial t}+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v)+\frac{\partial}{\partial z}(\rho u)\right)= -\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r j_{k,r}\right)+\frac{\partial j_{k,z}}{\partial z}\right)+\dot{\omega}_k \tag{2.46}$$

The second terms in Equations (2.43)-(2.46) are actually the continuity equation (Equation (2.29)) multiplied by the dependent variable in the corresponding equation,

and hence become zero. Therefore, Equations (2.43)-(2.46) can be written as;

r-momentum;

$$\rho\left(\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial r}+u\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial r}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rr})+\frac{\partial \tau_{zr}}{\partial z}-\frac{1}{r}\tau_{\theta\theta}\right)+\rho g_r, \quad (2.47)$$

z-momentum;

$$\rho\left(\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial r}+u\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial z}-\left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rz})+\frac{\partial \tau_{zz}}{\partial z}\right)+\rho g_z, \quad (2.48)$$

energy;

$$\begin{aligned}\rho\hat{C}_p\left(\frac{\partial T}{\partial t}+v\frac{\partial T}{\partial r}+u\frac{\partial T}{\partial z}\right) &= \left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda\frac{\partial T}{\partial r}\right)+\frac{\partial}{\partial z}\left(\lambda\frac{\partial T}{\partial z}\right)\right] \\ &- \sum_{k=1}^{N}\hat{C}_{p,k}\left(j_{k,r}\frac{\partial T}{\partial r}+j_{k,z}\frac{\partial T}{\partial z}\right) \\ &- \sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k \\ &- \nabla\cdot\mathbf{q_R},\end{aligned} \quad (2.49)$$

species;

$$\rho\left(\frac{\partial Y_k}{\partial t}+v\frac{\partial Y_k}{\partial r}+u\frac{\partial Y_k}{\partial z}\right)=-\left(\frac{1}{r}\frac{\partial}{\partial r}(rj_{k,r})+\frac{\partial j_{k,z}}{\partial z}\right)+\dot{\omega}_k. \quad (2.50)$$

Substituting the stress tensor expressions into Eqs. (2.47-2.48) and reaaranging

r-momentum;

$$\begin{aligned}\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial r}+u\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial r} \\ &+ \nu\left(\frac{4}{3}\frac{\partial^2 v}{\partial r^2}+\frac{4}{3r}\frac{\partial v}{\partial r}-\frac{4}{3}\frac{v}{r^2}+\frac{\partial^2 v}{\partial z^2}\right)+\frac{\nu}{3}\left(\frac{\partial^2 u}{\partial r\partial z}\right) \\ &+ \frac{1}{\rho}\left(\frac{4}{3}\frac{\partial v}{\partial r}-\frac{2}{3}\frac{\partial u}{\partial z}-\frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial r}+\frac{1}{\rho}\left(\frac{\partial u}{\partial r}+\frac{\partial v}{\partial z}\right)\frac{\partial \mu}{\partial z}+g_r\end{aligned} \quad (2.51)$$

z-momentum;

$$\begin{aligned}\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial z}+v\frac{\partial u}{\partial r} &= -\frac{1}{\rho}\frac{\partial p}{\partial z} \\ &+ \nu\left(\frac{4}{3}\frac{\partial^2 u}{\partial z^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{\partial^2 u}{\partial r^2}\right)+\frac{\nu}{3}\left(\frac{\partial^2 v}{\partial r\partial z}+\frac{1}{r}\frac{\partial v}{\partial z}\right) \\ &+ \frac{1}{\rho}\left(\frac{4}{3}\frac{\partial u}{\partial z}-\frac{2}{3}\frac{\partial v}{\partial r}-\frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial z}+\frac{1}{\rho}\left(\frac{\partial u}{\partial r}+\frac{\partial v}{\partial z}\right)\frac{\partial \mu}{\partial r}+g_z.\end{aligned} \quad (2.52)$$

Similar treatment of the derivatives in energy and species equations (Equations (2.49) and (2.50)) and division by $\rho\hat{C}_p$ and ρ, respectively lead to the final form of these equations;

energy;

$$\begin{aligned}\frac{\partial T}{\partial t}+v\frac{\partial T}{\partial r}+u\frac{\partial T}{\partial z} &= \frac{\lambda}{\rho\hat{C}_p}\left(\frac{\partial^2 T}{\partial r^2}+\frac{1}{r}\frac{\partial T}{\partial r}+\frac{\partial^2 T}{\partial z^2}\right) \\ &+ \frac{1}{\rho\hat{C}_p}\left(\frac{\partial T}{\partial r}\right)\frac{\partial \lambda}{\partial r}+\frac{1}{\rho\hat{C}_p}\left(\frac{\partial T}{\partial z}\right)\frac{\partial \lambda}{\partial z} \\ &- \frac{1}{\rho\hat{C}_p}\sum_{k=1}^{N}\hat{C}_{p,k}\left(j_{k,r}\frac{\partial T}{\partial r}+j_{k,z}\frac{\partial T}{\partial z}\right) \\ &- \frac{1}{\rho\hat{C}_p}\sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k \\ &- \frac{1}{\rho\hat{C}_p}\nabla\cdot\mathbf{q}_R,\end{aligned} \tag{2.53}$$

species;

$$\frac{\partial Y_k}{\partial t}+v\frac{\partial Y_k}{\partial r}+u\frac{\partial Y_k}{\partial z}=-\frac{1}{\rho}\left(\frac{\partial j_{k,r}}{\partial r}+\frac{j_{k,r}}{r}+\frac{\partial j_{k,z}}{\partial z}\right)+\frac{\dot{\omega}_k}{\rho}. \tag{2.54}$$

Spatial components of the mass–flux vector appearing in energy and species equations are determined by using

$$j_{k,z}=-\rho D_{km}\frac{\partial Y_k}{\partial z}, \tag{2.55}$$

$$j_{k,r}=-\rho D_{km}\frac{\partial Y_k}{\partial r}. \tag{2.56}$$

2.3 Equation Summary

The governing equations to be solved for the simulation of two-dimensional, axi-symmetric, laminar diffusion flame are summarized below. In order to surmount the stability problem associated with the discretization of convective terms, the convective derivatives on the left-hand-side and the first-order derivatives on the right-hand-side are rearranged to yield

continuity;

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial z}(\rho u)+\frac{\rho v}{r}+\frac{\partial}{\partial r}(\rho v)=0 \tag{2.57}$$

r-momentum;

$$\begin{aligned}\frac{\partial v}{\partial t} = & - \grave{v}\frac{\partial v}{\partial r} - \grave{u}\frac{\partial v}{\partial z} - \frac{1}{\rho}\frac{\partial p}{\partial r} \\ & + \nu\left(\frac{4}{3}\frac{\partial^2 v}{\partial r^2} - \frac{4}{3}\frac{v}{r^2} + \frac{\partial^2 v}{\partial z^2}\right) + \frac{\nu}{3}\left(\frac{\partial^2 u}{\partial r \partial z}\right) \\ & + \frac{1}{\rho}\left(-\frac{2}{3}\frac{\partial u}{\partial z} - \frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial r} + \frac{1}{\rho}\left(\frac{\partial u}{\partial r}\right)\frac{\partial \mu}{\partial z} + g_r\end{aligned} \tag{2.58}$$

where

$$\grave{u} = u - \frac{1}{\rho}\frac{\partial \mu}{\partial z} \tag{2.59}$$

$$\grave{v} = v - \frac{4\nu}{3r} - \frac{4}{3\rho}\frac{\partial \mu}{\partial r} \tag{2.60}$$

z-momentum;

$$\begin{aligned}\frac{\partial u}{\partial t} = & - \acute{u}\frac{\partial u}{\partial z} - \acute{v}\frac{\partial u}{\partial r} - \frac{1}{\rho}\frac{\partial p}{\partial z} \\ & + \nu\left(\frac{4}{3}\frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 u}{\partial r^2}\right) + \frac{\nu}{3}\left(\frac{\partial^2 v}{\partial r \partial z} + \frac{1}{r}\frac{\partial v}{\partial z}\right) \\ & + \frac{1}{\rho}\left(-\frac{2}{3}\frac{\partial v}{\partial r} - \frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial z} + \frac{1}{\rho}\left(\frac{\partial v}{\partial z}\right)\frac{\partial \mu}{\partial r} + g_z\end{aligned} \tag{2.61}$$

where

$$\acute{u} = u - \frac{4}{3\rho}\frac{\partial \mu}{\partial z} \tag{2.62}$$

$$\acute{v} = v - \frac{\nu}{r} - \frac{1}{\rho}\frac{\partial \mu}{\partial r} \tag{2.63}$$

energy;

$$\begin{aligned}\frac{\partial T}{\partial t} = & - \tilde{u}\frac{\partial T}{\partial z} - \tilde{v}\frac{\partial T}{\partial r} \\ & + \frac{\lambda}{\rho \hat{C}_p}\left(\frac{\partial^2 T}{\partial r^2} + \frac{\partial^2 T}{\partial z^2}\right) - \frac{1}{\rho \hat{C}_p}\sum_{k=1}^{N}\hat{H}_k \dot{\omega}_k - \frac{1}{\rho \hat{C}_p}\nabla \cdot \mathbf{q_R}\end{aligned} \tag{2.64}$$

where

$$\tilde{u} = u - \frac{1}{\rho \hat{C}_p}\frac{\partial \lambda}{\partial z} + \frac{1}{\rho \hat{C}_p}\sum_{k=1}^{N} j_{k,z}\hat{C}_{p,k} \tag{2.65}$$

$$\tilde{v} = v - \frac{1}{\rho \hat{C}_p}\frac{\partial \lambda}{\partial r} - \frac{\lambda}{\rho \hat{C}_p}\frac{1}{r} + \frac{1}{\rho \hat{C}_p}\sum_{k=1}^{N} j_{k,r}\hat{C}_{p,k} \tag{2.66}$$

species;

$$\begin{aligned}\frac{\partial Y_k}{\partial t} = & - u\frac{\partial Y_k}{\partial z} - v\frac{\partial Y_k}{\partial r} \\ & - \frac{1}{\rho}\left(\frac{\partial j_{k,r}}{\partial r} + \frac{j_{k,r}}{r} + \frac{\partial j_{k,r}}{\partial z}\right) + \frac{\dot{\omega}_k}{\rho}.\end{aligned} \tag{2.67}$$

Mass fraction of N_2 and its diffusive fluxes are computed by

$$Y_N = 1 - \sum_{k=1}^{N-1} Y_k \tag{2.68}$$

and

$$\mathbf{j}_N = \sum_{k=1}^{N-1} \mathbf{j}_k \tag{2.69}$$

2.4 Initial and Boundary Conditions

Specification of initial and boundary conditions constitutes an essential part of the solution of governing equations. For steady computations, the initial conditions are relatively unimportant. Usually, they may consist of large amplitude perturbations superposed on a realistic mean flow or of a fully developed flow in a similar configuation as the problem under consideration. Typically, the solution is allowed to develop in time until a steady state is reached. For unsteady computations however, more care should be addressed while assigning the initial conditions and physically correct conditions should be provided.

In two-dimensional simulations, boundary conditions consist of wall (solid surface), symmetry, inflow and outflow conditions.

For most applications no slip velocity boundary conditions are used at solid surfaces for velocities. This condition is based on the assumption that there is no motion between the solid surface and the fluid. Implementation of this Drichlet-type boundary condition is simply by setting the velocity components equal to zero. Similarly, Drichlet type boundary condition is employed for temperature by imposing

the wall temperature on the fluid if the wall is isothermal. Unlike the velocities and temperature, Neumann-type boundary conditions are imposed on all species by setting species gradients normal to the wall equal to zero which is based on the fact that no species can diffuse through the walls (impermeability condition).

For applications where the solution domain is treated as symmetrical, there is no net flow across the symmetry axis. Therefore, the velocity component normal to this boundary is equal to zero. Moreover, as a consequence of symmetry, gradients of all dependent variables normal to the boundary vanish and are set to zero.

The values of the dependent variables at the inflow are known *a priori* and their implementation is straightforward for laminar flows. For the specification of outflow conditions, fully-developed boundary condition in which first-order streamwise derivatives are set to zero may be employed. In cases where the length of the physical system is not sufficient, so-called soft boundary condition [34, 38, 43] where second-order streamwise derivatives are set to zero may be utilized.

In order to present the reader a complete picture, the initial and boundary conditions considered in the present study are summarized in Table 2.1. Their implementation on the governing equations are described next.

Table 2.1: Initial & Boundary Conditions

IC	@ t=0,	$\forall z \wedge \forall r$:	$u = 0,$	$v = 0,$	$T = T_{ref},$	$Y_k = Y_{k,o}$
BC1	@ the center,	$\forall z \wedge \forall t$:	$\frac{\partial u}{\partial r} = 0,$	$v = 0,$	$\frac{\partial T}{\partial r} = 0,$	$\frac{\partial Y_k}{\partial r} = 0$
BC2	@ the wall ,	$\forall z \wedge \forall t$:	$u = 0,$	$v = 0,$	$T = T_{wall},$	$\frac{\partial Y_k}{\partial r} = 0$
BC3	@ the inlet,	$\forall r \wedge \forall t$:	$u = u_{in},$	$v = 0,$	$T = T_{in},$	$Y_k = Y_{k,in}$
BC4	@ the outlet,	$\forall r \wedge \forall t$:	$\frac{\partial u}{\partial z} = 0,$	$\frac{\partial v}{\partial z} = 0,$	$\frac{\partial T}{\partial z} = 0,$	$j_{k,z} = 0$

Case 1: *Centerline Boundary Condition*

Boundary conditions to be applied along the symmetry axis are shown in Table 2.1. Derivatives of the transport and thermodynamic properties with respect to the

symmetry axis are also considered to be zero. The governing equations contain singularity at the centerline which can be circumvented by applying L'Hôpital's rule [51], i.e,

$$\lim_{r\to 0}\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}=2\frac{\partial^2 u}{\partial r^2}, \tag{2.70}$$

$$\lim_{r\to 0}\frac{\partial^2 T}{\partial r^2}+\frac{1}{r}\frac{\partial T}{\partial r}=2\frac{\partial^2 T}{\partial r^2}, \tag{2.71}$$

$$\lim_{r\to 0}\frac{\partial j_{k,r}}{\partial r}+\frac{j_{k,r}}{r}=2\frac{\partial j_{k,r}}{\partial r}. \tag{2.72}$$

The set of equations to be solved along the centerline as a result of application of the boundary conditions are;

z-momentum;

$$\begin{aligned}\frac{\partial u}{\partial t}+\acute{u}\frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial z}\\ &+ \nu\left(\frac{4}{3}\frac{\partial^2 u}{\partial z^2}+2\frac{\partial^2 u}{\partial r^2}\right)+\frac{2\nu}{3}\left(\frac{\partial^2 v}{\partial r\partial z}\right)\\ &+ \frac{1}{\rho}\left(-\frac{4}{3}\frac{\partial v}{\partial r}\right)\frac{\partial \mu}{\partial z}+g_z\end{aligned} \tag{2.73}$$

energy;

$$\frac{\partial T}{\partial t}+\tilde{u}\frac{\partial T}{\partial z}=\frac{\lambda}{\rho\hat{C}_p}\left(2\frac{\partial^2 T}{\partial r^2}+\frac{\partial^2 T}{\partial z^2}\right)-\frac{1}{\rho\hat{C}_p}\sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k-\frac{1}{\rho\hat{C}_p}\nabla\cdot\mathbf{q_R} \tag{2.74}$$

species;

$$\frac{\partial Y_k}{\partial t}+u\frac{\partial Y_k}{\partial z}+v\frac{\partial Y_k}{\partial r}=-\frac{1}{\rho}\left(2\frac{\partial j_{k,r}}{\partial r}+\frac{\partial j_{k,r}}{\partial z}\right)+\frac{\dot{\omega}_k}{\rho} \tag{2.75}$$

Case 2: *Exit Boundary Condition*

Fully developed conditions are imposed at the outflow in which the first-order streamwise derivatives are set to zero. Instead of dropping relevant terms from the governing equations, application of this condition is carried out by assigning the value of the dependent variable at the outflow to its value at the neighboring grid point, i.e.,

$$\phi|_{NZ}=\phi|_{NZ-1} \tag{2.76}$$

Case 3: *Wall Boundary Condition*

At the wall, Drichlet-type boundary conditions are employed for velocities and temperature whereas Neumann-type boundary conditions are utilized for species as shown in Table 2.1. Implementation of the Drichlet-type boundary conditions is performed by excluding the governing equations from the solution at the wall. Neumann-type boundary condition for species is applied by setting species wall concentration equal to concentration at the neighboring grid point, i.e.,

$$Y_k|_N R = Y_k|_{NR-1} \tag{2.77}$$

2.5 Radiative Transfer Equation

The radiative energy transport is governed by radiative transfer equation which can be written as

$$\frac{dI}{ds} = (\Omega \cdot \nabla) I(\mathbf{r}, \Omega) = -\kappa I(\mathbf{r}, \Omega) + \kappa I_b(\mathbf{r}) \tag{2.78}$$

where $I(\mathbf{r}, \Omega)$ is the radiation intensity at a position $\mathbf{r}$ in the direction Ω, κ is the absorption coefficient of the medium and I_b is the blackbody radiation intensity at the temperature of the medium. The expression on the left-hand-side is the gradient of the intensity in the specified direction Ω. The two terms on the right-hand-side represent the changes in the intensity due to absorption and emission, respectively.

In axisymmetric cylindrical coordinates, Equation (2.78) takes the following form

$$\frac{dI}{ds} = \frac{\mu}{r}\frac{\partial}{\partial r}(rI) - \frac{1}{r}\frac{\partial}{\partial \phi}(\eta I) + \xi \frac{dI}{dz} = -\kappa I + \kappa I_b \tag{2.79}$$

where $\mu (\equiv \sin\theta \cos\phi)$, $\eta (\equiv \sin\theta \sin\phi)$ and $\xi (\equiv \cos\theta)$ are the direction cosines in r, θ and z directions, respectively (see Figure 2.1).

If the surfaces bounding the medium are diffuse and gray at a specified temperature, Equation (2.78) is subject to following boundary condition

$$I(\mathbf{r_w}, \Omega) = \varepsilon_w I_b + \frac{(1-\varepsilon_w)}{\pi} \int_{\mathbf{n_w}\cdot\Omega'<0} \left|\mathbf{n_w} \cdot \Omega'\right| I(\mathbf{r_w}, \Omega') d\Omega' \tag{2.80}$$

where $I(\mathbf{r_w}, \Omega)$ and $I(\mathbf{r_w}, \Omega')$ are the radiative intensities leaving and incident on the surface at a boundary location, ε_w is the surface emissivity, I_b is the blackbody

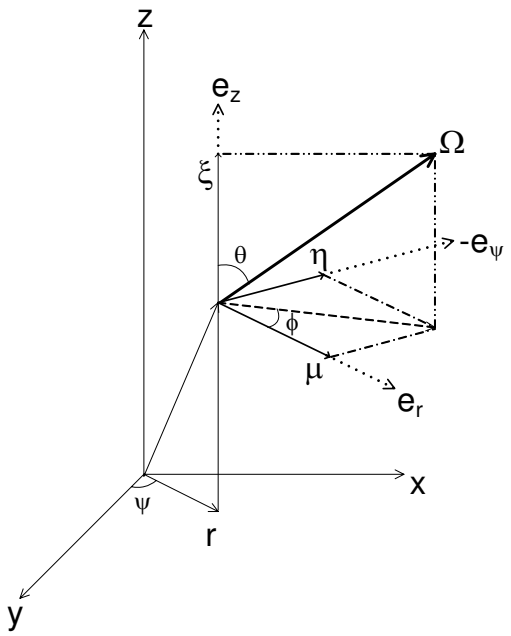


Figure 2.1: Cylindrical space-angle coordinate system in three dimensions.

radiation intensity at the surface temperature, **n** is the unit normal vector and $n_w \cdot \Omega'$ is the cosine of the angle between incoming direction Ω' and the surface normal. The first and second terms on the right-hand-side of Equation (2.80) stand for the contribution to the leaving intensity due to emission from the surface and reflection of incoming radiation, respectively.

2.6 Methane-Air Chemical Reaction Mechanism

Methane-air reaction mechanisms with varying complexity, involving a few steps to as many as several hundreds, are available in the literature. By making use of an appropriate reaction mechanism, the mass production rate, $\dot{\omega}_k$ for the k^{th} species appearing in energy and species equations can be evaluated.

For a total M number of reversible elementary reactions with N chemical species, the general form of a single arbitrary chemical reaction may be written as

$$\sum_{k=1}^{N} \nu_{jk}^{R} X_k \rightleftharpoons \sum_{k=1}^{N} \nu_{jk}^{P} X_k, \qquad j = 1,2,...,M, \tag{2.81}$$

where ν_{jk}^{R} and ν_{jk}^{P} are the stoichiometric coefficients on the reactants and products

side of the equation, respectively, for the kth species in the jth reaction and X_k represents the chemical symbol for the kth species. The net production rate $\dot{\omega}_k$ for the kth species is the summation of the rate of progress for all reactions involving the kth species:

$$\dot{\omega}_k = \sum_{j=1}^{M} (\nu_{jk}^P - \nu_{jk}^R) R_j. \tag{2.82}$$

The rate of progress, R_j, for jth reaction is given by the difference of the forward rates minus the reverse rates:

$$R_j = k_{fj} \prod_{k=1}^{N} [X_k]^{\nu_{jk}^R} - k_{bj} \prod_{k=1}^{N} [X_k]^{\nu_{jk}^P}. \tag{2.83}$$

where $[X_k]$ denotes the molar concentration of the kth species in the mixture and k_{fj} and k_{bj} are the rate constants for the forward and reverse rate constants of the jth reaction, respectively. The forward rate constants are expressed by the following semi-empirical relation based on Arrhenius law:

$$k_{fj} = A_j T^{\beta_j} exp\left(\frac{-E_{a,j}}{RT}\right). \tag{2.84}$$

where A_j, β_j and $E_{a,j}$ are the pre-exponential factor, the temperature exponent and the activation energy, respectively. These three parameters are usually determined from the experiments.

The reverse rate constants are usually expressed in terms of the forward rate constants and the equilibrium constants, $K_{e,j}$, as

$$k_{bj} = \frac{k_{fj}}{K_{e,j}}. \tag{2.85}$$

Equilibrium constants are determined by thermodynamic measurements and calculations.

Detailed simulation of reacting flows relies on complete reaction mechanisms. However, such complete mechanisms are too complex to be used in detailed multi-dimensional computational studies. Considerable amount of computational time is required to calculate the chemical species concentrations involved in detailed

chemistry. Furthermore, incorporation of multi-step chemistry necessitates the solution of a large set of extremely stiff equations due to the fact that each of the component reactions occurs at vastly different time scales. Therefore it is desirable to reduce the total number species and chemical reaction steps in order to keep the computations manageable.

Global (one-step) mechanisms are sometimes used to specify the overall kinetics. The mechanisms represent the stoichiometric relations among the major species including fuel, oxidizer and the most stable combustion products. Basic information related to flow and chemistry, such as peak temperature and major species profiles can be obtained with relative accuracy with these mechanisms.

Considering the fact that, the present investigations constitutes the initial steps towards the simulation of turbulent reacting flows and its objective is to demonstrated the capability of the developed algorithm to handle flows with chemical reaction and radiation, a global (one-step) reaction mechanism will be employed in the computations.

2.6.1 One-Step Reaction Mechanism

The one-step reaction mechanism for methane-air combustion in the presence of an inert gas (N_2) considered in this study can be shown as

$$CH_4 + 2O_2 + N_2 \rightarrow 2H_2O + CO_2 + N_2. \tag{2.86}$$

In a study by Tarhan [1], the predictive performances of three different one-step models given by Khalil *et al.* [52], Wesbrook and Dryer [53], Hsu and Mahalingam [54] were tested on the methane-air diffusion problem [45] under consideration. Based on the numerical experimentation, it was concluded that the predictions obtained with the model proposed by Khalil *et al.* [52] showed the best agreement with the experimental measurements. According to this model, rate of fuel consumption can be represented by the following expression;

$$-\dot{\omega}_{CH_4} = \rho^2 Y_{CH_4} Y_{O_2} A \exp(-E_a/RT) \tag{2.87}$$

where $A = 10^{13}$ cm^3/g·s and $E_a/R = 1.84 \times 10^4 K$.

Following Khalil *et al.* [52] and Tarhan [1], the rate of fuel consumption will be evaluated using Equation (2.87) in the present investigation.

2.7 Transport and Thermodynamic Properties

One of the major tasks in the simulation of reacting flows is the evaluation of transport and thermodynamic coefficients of the species and the mixture which are non-linear functions of local composition, temperature and pressure of the mixture. This evaluation is highly computationally intensive and reserves expensive part of the computations. In the present study, CHEMKIN (Chemical Kinetics) [55] and TRANSPORT [56] packages which are ready-to-use FORTRAN codes are utilized for the evaluation of transport and thermodynamic coefficients. The CHEMKIN Gas-Phase Subroutine library provides equation of state variables, thermodynamic properties and the chemical production rates while the TRANSPORT packages evaluates the transport properties, such as the viscosities, the thermal conductivities, and the diffusion coefficients. The details of the models used in these packages are described below.

The general equation of state of an ideal, multi-species gas is used in CHEMKIN for the calculation of mass density defined by

$$\rho = \frac{p\overline{W}}{RT}, \tag{2.88}$$

Mean molecular weight $\overline{W}$ can be calculated by

$$\overline{W} = \frac{1}{\sum_{k=1}^{N} Y_k/W_k}, \tag{2.89}$$

CHEMKIN presumes that standard-state thermodynamic properties are thermally perfect, in that they are only functions of temperature, and are given in terms of polynomial fits to the molar heat capacities at constant pressure:

$$\frac{C_{p,k}^o}{R} = a_{0k} + a_{1k}T_k + a_{2k}T_k^2 + a_{3k}T_k^3 + a_{4k}T_k^4. \tag{2.90}$$

where superscript o refers to standard-state one atmosphere. For ideal gases, the heat capacities are independent of pressure, and the standard-state values are the actual

ones. Another thermodynamic property, molar enthalpy is evaluated by

$$H_k^o = \int_{298}^{T_k} C_{p,k}^o dT + H_k^o(298). \tag{2.91}$$

In the present study, mass units are utilized and hence these properties are converted to mass units as

$$\hat{C}_{p,k} = \frac{C_{p,k}}{W_k} \tag{2.92}$$

and

$$\hat{H}_k = \frac{H_k}{W_k} \tag{2.93}$$

The mixture-averaged specific heats are evaluated by

$$\hat{C}_p = \frac{1}{W_k} \sum_{k=1}^{N} C_{p,k} X_k \tag{2.94}$$

CHEMKIN package uses NASA thermodynamic data base [57] that contains polynomial fits to thermodynamic properties of many species involved in combustion and other applications.

TRANSPORT package provides pure species transport properties as well as multi-component gas mixture properties. In the TRANSPORT package, evaluation of single component viscosities μ_k and binary diffusion coefficients D_{jk} are based on standard kinetic theory expressions which can be found in [56, 58]. TRANSPORT package evaluates gas mixture properties from these pure species properties by certain mixture averaging rules [56, 58]. The mixture averaged viscosity is determined by

$$\mu = \sum_{k=1}^{N} \frac{X_k \mu_k}{\sum_{j=1}^{N} X_j \Phi_{kj}} \tag{2.95}$$

where,

$$\Phi_{kj} = \frac{1}{\sqrt{8}} \left(1 + \frac{W_k}{W_j}\right)^{-1/2} \left[1 + \left(\frac{\mu_k}{\mu_j}\right)^{1/2} \left(\frac{W_k}{W_j}\right)^{1/4}\right]^2. \tag{2.96}$$

The mixture diffusion coefficient for species k is computed as

$$D_{km} = \frac{1 - Y_k}{\sum_{j \neq k}^{N} X_j / D_{jk}}. \tag{2.97}$$

The mixture-averaged thermal conductivity λ is evaluated using an averaging formula that combines the values of individual pure species, λ_k [56]:

$$\lambda = \frac{1}{2}\left[\sum_{k=1}^{N} X_k \lambda_k + \left(\sum_{k=1}^{N} \frac{X_k}{\lambda_k}\right)^{(-1)}\right]. \tag{2.98}$$

CHAPTER 3

METHODOLOGY

3.1 Preamble

As mentioned earlier, the priciple objective of this study is development of an operator-splitting based algorithm, similar to PISO, for the computation of transient reacting radiating flows. Different from its predecessor, the present method operates on non-staggered grid topology and employs implicit time integration at all splitting stages using a semi-discrete approach also known as numerical Method of Lines (MOL) [59]. The radiative heat transfer computations will be carried out by means of incorporation of previously developed radiation codes for gray [49] and non-gray media [50, 60].

In the proceeding sections, first, an overview of the splitting procedure followed by the description of the semi-discrete approach (MOL) will be presented. The application of procedure to reacting radiating flows using semi-discrete formulation of the governing equations will be explained next.

3.2 Overview of the Splitting Procedure

The developed algorithm is based on splitting the solution of conservation equations into a series of predictor-corrector stages (see Figure (3.1)). Each stage can be regarded as a block: within a block, conservation equations of mass, momentum and energy are solved depending on the type of flow field. For instance, if the flow under consideration is incompressible, blocks consist of the solution of momentum equations only or if it is reacting, all conservation equations are solved within a block.

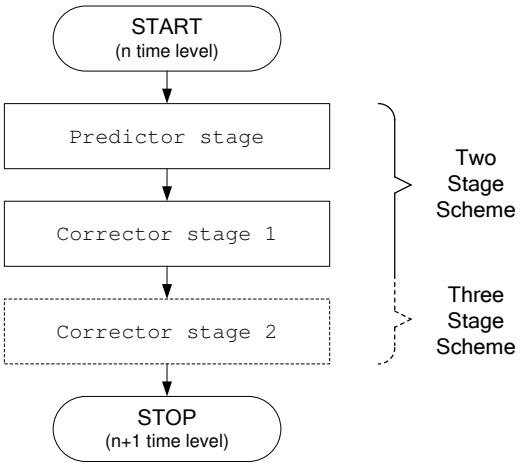


Figure 3.1: Overview of the splitting procedure.

Number of corrector stages succeeding the predictor stage depends on the desired level of accuracy (one at least).

This modular nature of the scheme made its gradual development possible in the course of the present investigation such that the algorithm was first developed for isothermal incompressible flows and then it was extended to non-isothermal flows by the incorporation of the solution of the energy equation. Finally, the algorithm was extended to reacting radiating flows by the incorporation of the solution species and radiative transfer equations.

3.3 Semi-Discrete Approach

In this study, the governing equations are solved using the numerical Method of Lines which a is semi-discrete technique for the solution of time-dependent partial differential equations (PDEs). The MOL treats a general PDE of the form

$$\frac{\partial \phi}{\partial t} = f(r,z) \tag{3.1}$$

in two stages. First, the dependent variables are kept continuous in time and the PDE is discretized in space on a dimension by dimension basis using any readily

available spatial discretization packages such as finite-difference, finite-element or finite-volume based schemes. This leads to a set of ordinary differential equations (ODEs) which can be represented by the following equation

$$\frac{d}{dt}\left(\phi_{i,j}\right) \approx f_{i,j} \qquad i = 1,2,\ldots,NR \quad ; \quad j = 1,2,\ldots,NZ \tag{3.2}$$

where the ordinary derivative which replaces the partial derivative in Equation (3.1) represents the time variation of the value of the dependent variable ϕ at a discrete point (i, j), $f_{i,j}$ is the right-hand-side function computed at the same discrete point, NR and NZ are the number of grid points in r and z directions as a result of discretization of the solution domain, respectively. Next, the ODEs are integrated in time using a sophisticated ODE solver which takes the burden of time-discretization and maintains the accuracy and stability of the evolving solution.

By this way, MOL not only offers the simplicity of the explicit methods but also the superiority (stability advantage) of the implicit ones unless a poor numerical scheme for the solution of ODEs. The advantages of the MOL approach are two-folds. First, it is possible to use higher-order approximations for the discretization of spatial derivatives without significant increase in computational complexity. Second, the utilization of highly efficient and reliable ODE solvers means that comparable order of accuracy can also be achieved in the time integration without using extremely small time steps.

In consideration of the MOL solution, the governing equations will be presented in their semi-discrete forms for sections to come. Integration of the semi-discrete equations using an implicit ODE solver will be explained in Chapter 4.

3.4 Splitting Procedure for Reacting Radiating Flows

A three stage scheme in which there are one predictor and two corrector stages for the solution of momentum, energy and species equations will be presented (see Figure (3.2)). In order to account for the rapid variations in temperature and concentration over a time-step due to combustion, the sequence of solution commences with the energy equation and is then followed by species and momentum

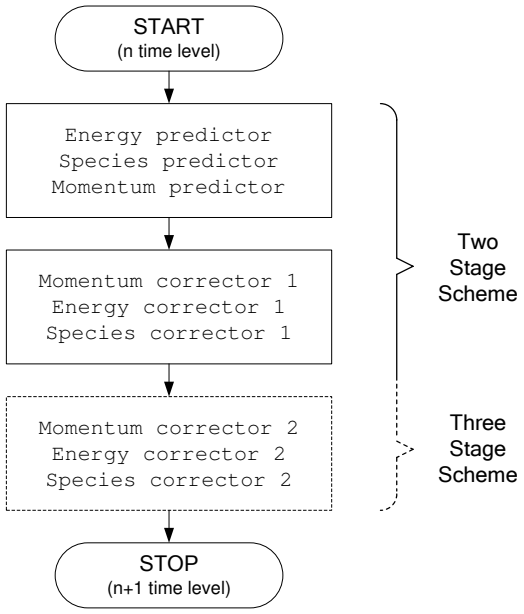


Figure 3.2: Overview of the splitting procedure for reacting radiating flows.

equations as suggested by [19]. In the course of splitting; temperature, pressure and concentration dependent physical properties are updated at the start of every stage using CHEMKIN-III [55] and TRANSPORT [56] packages; radiative source term to be used in the energy equation is computed before attempting the solution of energy equation in each stage by the help of radiation module.

Following the semi-discrete approach, the conservation equations at each splitting phase can be shown as

$$\frac{d\phi}{dt} \approx Residual \tag{3.3}$$

where in this case ϕ is one of the dependent variables v, u, T or Y_K and *Residual* is finite difference representation of the terms on the right hand side of the parent equations (Equations (2.58), (2.61), (2.64) and (2.67)). According to this notion, the residual expressions for momentum, energy and species equations are;

r-momentum;

$$\begin{aligned} Residual_{m_r} = FD\{ \quad & - \quad \grave{v}\frac{\partial v}{\partial r} - \grave{u}\frac{\partial v}{\partial z} - \frac{1}{\rho}\frac{\partial p}{\partial r} \\ & + \quad \nu\left(\frac{4}{3}\frac{\partial^2 v}{\partial r^2} - \frac{4}{3}\frac{v}{r^2} + \frac{\partial^2 v}{\partial z^2}\right) + \frac{\nu}{3}\left(\frac{\partial^2 u}{\partial r \partial z}\right) \\ & + \quad \frac{1}{\rho}\left(-\frac{2}{3}\frac{\partial u}{\partial z} - \frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial r} + \frac{1}{\rho}\left(\frac{\partial u}{\partial r}\right)\frac{\partial \mu}{\partial z} + g_r\}, \end{aligned} \tag{3.4}$$

z-momentum;

$$\begin{aligned} Residual_{m_z} = FD\{ \quad & - \quad \acute{u}\frac{\partial u}{\partial z} - \acute{v}\frac{\partial u}{\partial r} - \frac{1}{\rho}\frac{\partial p}{\partial z} \\ & + \quad \nu\left(\frac{4}{3}\frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 u}{\partial r^2}\right) + \frac{\nu}{3}\left(\frac{\partial^2 v}{\partial r \partial z} + \frac{1}{r}\frac{\partial v}{\partial z}\right) \\ & + \quad \frac{1}{\rho}\left(-\frac{2}{3}\frac{\partial v}{\partial r} - \frac{2}{3}\frac{v}{r}\right)\frac{\partial \mu}{\partial z} + \frac{1}{\rho}\left(\frac{\partial v}{\partial z}\right)\frac{\partial \mu}{\partial r} + g_z\}, \end{aligned} \tag{3.5}$$

energy;

$$\begin{aligned} Residual_e = FD\{ \quad & - \quad \tilde{u}\frac{\partial T}{\partial z} - \tilde{v}\frac{\partial T}{\partial r} \\ & + \quad \frac{\lambda}{\rho\hat{C}_p}\left(\frac{\partial^2 T}{\partial r^2} + \frac{\partial^2 T}{\partial z^2}\right) - \frac{1}{\rho\hat{C}_p}\sum_{k=1}^{N}\hat{H}_k\dot{\omega}_k - \frac{1}{\rho\hat{C}_p}\nabla\cdot\mathbf{q_R}\}, \end{aligned} \tag{3.6}$$

species;

$$\begin{aligned} Residual_{s_k} = FD\{ \quad & - \quad u\frac{\partial Y_k}{\partial z} - v\frac{\partial Y_k}{\partial r} \\ & - \quad \frac{1}{\rho}\left(\frac{\partial j_{k,r}}{\partial r} + \frac{j_{k,r}}{r} + \frac{\partial j_{k,r}}{\partial z}\right) + \frac{\dot{\omega}_k}{\rho}\} \qquad k = 1,\ldots,N. \end{aligned} \tag{3.7}$$

where the subscripts m_r, m_z, e and s_k denote that the residual expression belongs to the r-momentum, z-momentum, energy and species equations, respectively and $FD\{\}$ is a symbolic operator which simply represents the finite difference expressions in a compact form (details of which will be dealt with later). The splitting procedure is demonstrated below by letting the superscript n denote present time level and *, **, *** denote the intermediate values between the n and $n+1$ levels.

(a) Predictor Stage:

The equations to be solved for this stage are:

(i) Energy predictor:

$$\frac{dT}{dt} \approx Residual_e^n \tag{3.8}$$

(ii) Species predictor:

$$\frac{dY_k}{dt} \approx Residual_{s_k}^n \qquad k = 1,\ldots,N \tag{3.9}$$

(iii) Momentum predictor:

$$\frac{dv}{dt} \approx Residual_{m_r}^n \tag{3.10}$$

and

$$\frac{du}{dt} \approx Residual_{m_z}^n \tag{3.11}$$

The solution of Eqs. (3.8-3.11) yields the first intermediate fields, T^*, Y_k^*, v^* and u^*. It should be noted that velocity field obtained as a result of the predictor stage does not necessarily satisfy equation of continuity since it has been obtained using a guessed (initial) pressure field.

(b) First Corrector Stage:

(i) Momentum corrector: A new velocity field v^{**}, u^{**} together with the corresponding pressure field p^* are now sought which will satisfy the semi-discrete form of the equation of continuity for this stage

$$\frac{\rho^* - \rho^n}{\Delta t} \approx -\left(\frac{\partial}{\partial r}(\rho^* v^{**}) + \frac{\rho^* v^{**}}{r} + \frac{\partial}{\partial z}(\rho^* u^{**})\right). \tag{3.12}$$

For this purpose, a pressure equation is derived by taking the divergence of momentum equations (See Appendix for details) as follows;

$$\left(\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2}\right)^* + LHS^* = -\frac{\partial}{\partial t}\left(\frac{\partial}{\partial r}(\rho v) + \frac{\rho v}{r} + \frac{\partial}{\partial z}(\rho u)\right) + RHS^* \tag{3.13}$$

where

$$\begin{aligned} LHS &= \frac{\partial^2}{\partial z^2}(\rho^n u^* u^*) + \frac{\partial^2}{\partial r^2}(\rho^n v^* v^*) + \frac{2}{r}\frac{\partial}{\partial r}(\rho^n u^* v^*) \\ &+ \frac{2}{r}\frac{\partial}{\partial z}(\rho^n u^* v^*) + 2\frac{\partial^2}{\partial r \partial z}(\rho^n u^* v^*) \end{aligned} \tag{3.14}$$

$$RHS = -\left(\frac{\partial^2 \tau_{zz}}{\partial z^2} + \frac{\partial^2 \tau_{rr}}{\partial r^2} + 2\frac{\partial^2 \tau_{rz}}{\partial r \partial z} + \frac{2}{r}\frac{\partial \tau_{rr}}{\partial r} + \frac{2}{r}\frac{\partial \tau_{rz}}{\partial z} - \frac{1}{r}\frac{\partial \tau_{\theta\theta}}{\partial r}\right)^* \tag{3.15}$$

Using first order differences for the evaluation of time derivative in the right-hand-side of Equation (3.13) and rearranging, one obtains

$$\begin{aligned}\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* = &- \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^* v^{**})+\frac{\rho^* v^{**}}{r}+\frac{\partial}{\partial z}(\rho^* u^{**})\right)\\ &+ \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*)+\frac{\rho^n v^*}{r}+\frac{\partial}{\partial z}(\rho^n u^*)\right)\\ &- LHS^* + RHS^*.\end{aligned} \tag{3.16}$$

As can be noticed, Equation (3.16) is implicit due to the presence of time-advanced terms on the right-hand-side. To alleviate this, equation of continuity (Equation (3.12)) is substituted into Equation (3.16) yielding

$$\begin{aligned}\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* = &\frac{1}{\Delta t}\left(\frac{\rho^* - \rho^n}{\Delta t}\right)\\ &+ \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*)+\frac{\rho^n v^*}{r}+\frac{\partial}{\partial z}(\rho^n u^*)\right)\\ &- LHS^* + RHS^*\end{aligned} \tag{3.17}$$

in which the only time-advanced term left is ρ^*. This term can be eliminated in favor of p^* by the help of equation of the form

$$\rho^* = \frac{p^* \overline{W}^*}{RT^*} \tag{3.18}$$

which yields the final form of the pressure equation for this stage

$$\begin{aligned}\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* - \frac{1}{\Delta t^2}\frac{p^* \overline{W}^*}{RT^*} = &-\frac{\rho^n}{\Delta t^2}\\ &+ \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*)+\frac{\rho^n v^*}{r}+\frac{\partial}{\partial z}(\rho^n u^*)\right)\\ &- LHS^* + RHS^*.\end{aligned} \tag{3.19}$$

It should be noted that, by following the procedure described above, not only an explicit equation for the determination of pressure field is obtained (the only unknown in Equation (3.19) is p^*) but also the satisfaction of mass conservation constraint is automatically guaranteed by the enforcement of equation of continuity in the derivation of Equation (3.19). Once the corrected pressure field (p^*) is obtained

by the solution of Eq. (3.19), ρ^* is computed by invoking Eq. (3.18). Having obtained p^* and ρ^*, momentum equations for this stage

$$\frac{dv}{dt} \approx Residual^*_{m_r} \tag{3.20}$$

and

$$\frac{du}{dt} \approx Residual^*_{m_z} \tag{3.21}$$

are solved to yield the first corrected velocity fields, v^{**} and u^{**}.

(ii) Energy corrector: Energy equation to be solved for this stage is

$$\frac{dT}{dt} \approx Residual^*_e \tag{3.22}$$

solution of which results in the second intermediate temperature field T^{**}.

(iii) Species corrector: Species equation to be solved for this stage is

$$\frac{dY_k}{dt} \approx Residual^*_{s_k} \qquad\qquad k = 1, \ldots, N \tag{3.23}$$

soltuion of which finalizes the the first corrector stage yielding the second intermediate species field Y_k^{**}.

(c) Second Corrector Stage:

The second corrector stage is the same as first corrector stage except for the fact that all *Residual* expressions are computed at ** level. For the sake of completeness, these equations are given below.

(i) Momentum corrector: Similar to the one derived for the first corrector stage, the pressure equation for the second corrector stage is

$$\begin{aligned}\left(\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2}\right)^{**} - \frac{1}{\Delta t^2}\frac{p^{**}\overline{W}^{**}}{RT^{**}} &= -\frac{\rho^*}{\Delta t^2} \\ &+ \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^* v^{**}) + \frac{\rho^* v^{**}}{r} + \frac{\partial}{\partial z}(\rho^* u^*)\right) \\ &- LHS^{**} + RHS^{**}.\end{aligned} \tag{3.24}$$

Density and pressure are related to each other through the equation of state of the form

$$\rho^{**} = \frac{p^{**}\overline{W}^{**}}{RT^{**}} \tag{3.25}$$

Having obtained p^{**} and ρ^{**} by the solution of Equation (3.24)-(3.25), the momentum equations to be solved for the determination of velocity field are

$$\frac{dv}{dt} \approx Residual_{m_r}^{**} \tag{3.26}$$

and

$$\frac{du}{dt} \approx Residual_{m_z}^{**}. \tag{3.27}$$

(ii) Energy corrector: The energy equation to be solved for the second corrector stage is

$$\frac{dT}{dt} \approx Residual_{e}^{**}. \tag{3.28}$$

(iii) Species corrector: The species equation to be solved for the second corrector stage is

$$\frac{dY_k}{dt} \approx Residual_{s_k}^{**} \qquad\qquad k = 1, \ldots, N. \tag{3.29}$$

At the end of second corrector stage, third intermediate fields T^{***}, Y_k^{***}, v^{***} and u^{***} are obtained. This concludes the algorithm for one time step as the the governing equations are now integrated from n to $n+1$ time level. More corrector stages can be employed but the experience shows that even the utilization of second corrector stage does not bring significant improvement in the accuracy, a finding which was also confirmed by other researchers [12, 19].

CHAPTER 4

NUMERICAL SOLUTION TECHNIQUE

4.1 Preamble

In what preceeded, a non-iterative pressure based algorithm for the computation of reacting radiating flows was described. The algorithm is based on a sequence of predictor-corrector stages and uses a powerful semi-discrete approach, the MOL, for the numerical solution of conservation equations.

In this chapter, the components of the solution strategy such as grid structure, discretization of the spatial derivatives, time integration, numerical solution of the pressure equation and computation of radiative source term will be explained in detail.

4.2 Grid Structure

One of the advantages gained by compressible formulations is that the complications associated incompressible formulations resulting from utilization of regular grids (odd-even decoupling or checker-board oscillations) explained earlier do not occur in which the inclusion of density variation in the continuity equation (Equation (2.29)) would not allow the formation of any checkerboard pattern in the pressure field [61].

Therefore, based on the confidence gained by the compressible formulation employed in the present investigation, solution of governing equations are carried out on 2D regular finite-difference grids with uniform spacing . The number of grids to be used for the CFD code in each direction is set according to the following formula which is dictated by the multi-grid package (MUDPACK) employed for the solution

of pressure equation

$$N = A \times (2^{(B-1)}) + 1 \tag{4.1}$$

where A is a prime pre-exponential factor (preferably small prime integers such as 2,3 or 5 for efficient multi-grid cycling) and B is any integer exponent chosen by the user depending on the dimensions of the physical system. The radiation grids are established by coarsening the CFD grids in such a way that each grid point on the radiation mesh corresponds exactly to a grid point on the CFD mesh. The representative grids for CFD and radiation codes are illustrated in Figure 4.1.

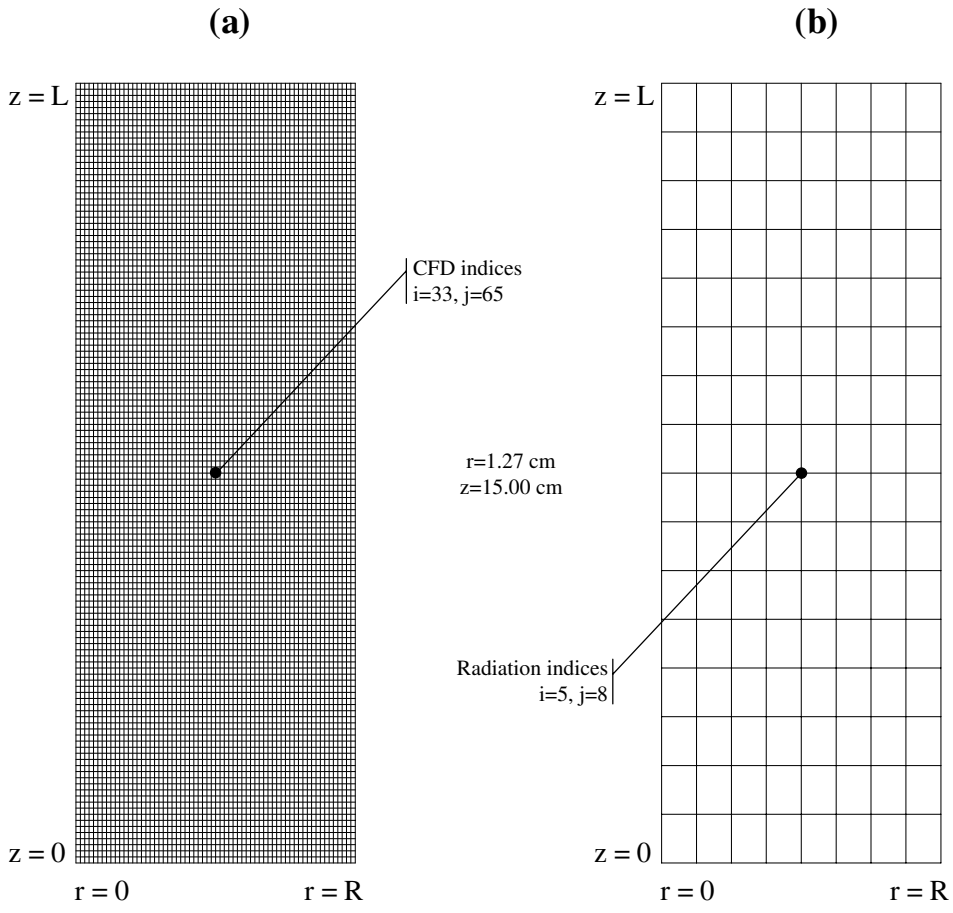


Figure 4.1: Representative grid structures for (a) CFD code; (b) Radiation code. i and j correspond to grid indices for r and z directions, respectively.

4.3 Computation of Spatial Derivatives

As mentioned previously, the first stage of the MOL solution consists of discretization of the spatial derivatives which converts the system of PDEs into an ODE initial value problem. MOL is flexible in the choice of discretization scheme to use. Any available algorithm and library routine for one-dimensional discretization can be applied to spatial derivatives in each direction on a dimension-by-dimension basis provided that the resulting system of ODEs is stable according to the linear stability theory [59]. The stability of the ODEs in convection dominated problems can only be maintained if the discretization scheme is consistent with the direction of propogation of information.

In view of this, Oymak and Selçuk [35] used a Lagrange interpolation polynomial based finite-difference scheme which can be written in compact form as follows

$$\frac{\partial u}{\partial x} = \sum_{i=1}^{n} b_i(x) u_i \tag{4.2}$$

where

$$b_i(x) = \frac{\sum_{\substack{k=1 \\ k\neq i}}^{n} \prod_{\substack{j=1 \\ j\neq i \\ j\neq k}}^{n} x - x_j}{\prod_{\substack{j=1 \\ j\neq i}}^{n} x_i - x_j}. \tag{4.3}$$

The fourth-order accurate (5-point) scheme used in [35] can be obtained by setting n equal to 5 in Equations (4.2) and (4.3) as

$$
\begin{aligned}
\left.\frac{\partial u}{\partial x}\right|_x &= \frac{\begin{array}{c}[(x-x_2)(x-x_3)(x-x_4)+(x-x_2)(x-x_3)(x-x_5)\\ +(x-x_2)(x-x_4)(x-x_5)+(x-x_3)(x-x_4)(x-x_5)]\end{array}}{(x_1-x_2)(x_1-x_3)(x_1-x_4)(x_1-x_5)}u_1 \\
&+ \frac{\begin{array}{c}[(x-x_1)(x-x_3)(x-x_4)+(x-x_1)(x-x_3)(x-x_5)\\ +(x-x_1)(x-x_4)(x-x_5)+(x-x_3)(x-x_4)(x-x_5)]\end{array}}{(x_2-x_1)(x_2-x_3)(x_2-x_4)(x_2-x_5)}u_2 \\
&+ \frac{\begin{array}{c}[(x-x_1)(x-x_2)(x-x_4)+(x-x_1)(x-x_2)(x-x_5)\\ +(x-x_1)(x-x_4)(x-x_5)+(x-x_2)(x-x_4)(x-x_5)]\end{array}}{(x_3-x_1)(x_3-x_2)(x_3-x_4)(x_3-x_5)}u_3 \\
&+ \frac{\begin{array}{c}[(x-x_1)(x-x_2)(x-x_3)+(x-x_1)(x-x_2)(x-x_5)\\ +(x-x_1)(x-x_3)(x-x_5)+(x-x_2)(x-x_3)(x-x_5)]\end{array}}{(x_4-x_1)(x_4-x_2)(x_4-x_3)(x_4-x_5)}u_4 \\
&+ \frac{\begin{array}{c}[(x-x_1)(x-x_2)(x-x_3)+(x-x_1)(x-x_2)(x-x_4)\\ +(x-x_1)(x-x_3)(x-x_4)+(x-x_2)(x-x_3)(x-x_4)]\end{array}}{(x_5-x_1)(x_5-x_2)(x_5-x_3)(x_5-x_4)}u_5. \qquad (4.4)
\end{aligned}
$$

Substitution of $x = x_1$, $x = x_2$, $x = x_3$, $x = x_4$ and $x = x_5$ into Equation (4.4) yields the fourth-order pure-downwind, biased-downwind, centered, biased-upwind and pure-upwind difference schemes based on Lagrange interpolation polynomials, respectively. Considering the fact that one-sided approximations (pure-downwind or pure-upwind) lead to undesired oscillations when used for the discretization of convective terms, biased-downwind or biased-upwind stencils were chosen for the first-order (convective) derivatives depending on the direction of the flow. Second-order derivatives were discretized using centered stencil by means of stage-wise differentiation of the first-order derivatives obtained from fourth-order centered scheme. This approach was successfully applied to the MOL solution of diverse range of fluid flow problems [38, 42, 43, 62].

In an attempt to apply the same principles to the simulation of chemically reacting flows, Tarhan [63] observed that spurious over- and under-shoots occur in the

vicinity of the steep velocity and temperature gradients The remedy proposed was the utilization of a Lagrange interpolation polynomial based difference scheme along with a total variation diminishing (TVD) flux limiter for the discretization of convective derivatives.

Here the reader should be reminded that, a numerical scheme is said to be TVD if the total variation does not increase in time. That is;

$$TV(u^{n+1}) \leq TV(u^n) \tag{4.5}$$

where $TV(u^n)$ is the total variation of the numerical solution at the time level t^n defined as

$$TV(u^n) = \sum_i |u_{i+1} - u_i| \tag{4.6}$$

and u_i stand for the approximate solutions at mesh nodes x_i.

For the satisfaction of the TVD condition, Van Leer flux limiter [64, 65] of the form

$$\Psi_i = \frac{r_i + |r_i|}{1 + |r_i|} \tag{4.7}$$

is inttroduced to second-order Lagrange interpolation based difference scheme (which can be obtained by setting n equal to 3 in Equations (4.2) and (4.3)) yielding the following upwind and downwind expressions, respectively:

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_i &= \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \\ &+ \frac{1}{x_i - x_{i-2}} \left[\Psi_i (u_i - u_{i-1}) - \Psi_{i-1} \frac{x_i - x_{i-1}}{x_{i-1} - x_{i-2}} (u_{i-1} - u_{i-2}) \right] \end{aligned} \tag{4.8}$$

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_i &= \frac{u_i - u_{i+1}}{x_i - x_{i+1}} \\ &+ \frac{1}{x_i - x_{i+2}} \left[\Psi_i (u_i - u_{i+1}) - \Psi_{i+1} \frac{x_i - x_{i+1}}{x_{i+1} - x_{i+2}} (u_{i+1} - u_{i+2}) \right]. \end{aligned} \tag{4.9}$$

In Equations (4.8) and (4.9), the limiters Ψ are function of ratios of consecutive variations given by

$$r_i = \frac{(u_{i+1} - u_i)/(x_{i+1} - x_i)}{(u_i - u_{i-1})/(x_i - x_{i-1})} \tag{4.10}$$

for upwind scheme and

$$r_i = \frac{(u_{i-1} - u_i)/(x_{i-1} - x_i)}{(u_i - u_{i+1})/(x_i - x_{i+1})} \tag{4.11}$$

for downwind scheme, respectively. The details and the derivation of the scheme be found elsewhere [1].

The discretization strategy used in the present investigation is based on the schemes developed and used by Oymak and Selçuk [35] and Tarhan [63]. In what follows, types of spatial derivatives encountered in the governing equations and the corresponding schemes utilized will be addressed.

4.3.1 Discretization of the Convective Terms

The discretization of convective terms requires particular attention in flows with recirculation zones where utilization of schemes which does not follow the direction of propogation leads to unstable solutions. In this study, this bottleneck is alleviated by constructing an adaptive spatial discretization scheme [35] which operates in a zone-dependent manner for the approximation of convective derivatives.The adaptive scheme works as follows. The code checks the signs of the coefficients ($\grave{u}$, $\grave{v}$, $\acute{u}$, $\acute{v}$, $\tilde{u}$, $\tilde{v}$, u, v) of the convective derivatives ($\frac{\partial v}{\partial z}$, $\frac{\partial v}{\partial r}$, $\frac{\partial u}{\partial z}$, $\frac{\partial u}{\partial r}$, $\frac{\partial T}{\partial z}$, $\frac{\partial T}{\partial r}$, $\frac{\partial Y_k}{\partial z}$, $\frac{\partial Y_k}{\partial r}$) and decides whether an upwind (Equation (4.8)) or downwind (Equation (4.9)) scheme is to be used. If the coefficient is positive, discretization of convective derivatives are carried out by an upwind scheme as the information is gathered from the upstream direction. If the coefficient is negative, a downwind scheme is used as the information regarding to the direction of propogation is recieved from downstream locations with respect to the point under consideration. Implementation of this procedure into the computer code is performed by casting the convective derivatives into following form

$$\left.\frac{\partial \phi}{\partial \bar{x}}\right|_i = \frac{C}{2}\left[(1-\varepsilon_C)\left.\frac{\partial \phi}{\partial \bar{x}}\right|_d + (1+\varepsilon_C)\left.\frac{\partial \phi}{\partial \bar{x}}\right|_u\right] \tag{4.12}$$

where ϕ is the pseudo one-dimensional dependent variable at any point along the $\bar{x}$ direction at any time. C denotes the coefficients of the convective derivatives. Subscripts d and u denote downwind and upwind stencils, respectively, and

$$\varepsilon_C = sign(C) = \frac{C}{|C|}. \tag{4.13}$$

4.3.2 Discretization of the Diffusive Terms

The diffusive terms in the governing equations appear as second-order derivatives of the dependent variables either with respect to one of the independent variables or both of them (in the form of mixed-derivatives). Since they correspond to diffusive effects, they are always centrally discretized [66]. The approach used in the present study for the evaluation of second-order derivatives is the stagewise differentiation of the first-order derivatives [35]. The first-order centered derivatives obatined by fourth-order centered scheme based on Lagrange interpolation polynomials (Equation (4.4)) are differentiated once more using the same scheme to obtain the necessary derivative.

4.3.3 Discretization of the Pressure Gradient Terms

Although pressure gradient term appears as a first-order derivative in equation of momentum (Equation (2.2)), its role is contribution to the momentum addition by molecular transport [58] and hence it has diffusive characteristics. Moreover, due to the elliptic nature of the equation for the determination of pressure (Equation (3.13)), any change in pressure can be felt in the complete solution domain immediately. Therefore, discretization of pressure gradient terms necessitates the utilization of centered schemes with relatively large stencil widths. In consideration of the abovelisted arguements, pressure gradients are approximated using fourth-order centered scheme (Equation (4.4)).

4.4 Time Integration

In Section 4.3, the first stage of the MOL solution, spatial discretization of the governing equations, was explained. In this section, the second stage, time integration, will be described in detail.

Recall that, substitution of the finite-difference approximations of spatial derivatives into the governing equations results in a system of ODEs of the form

$$\frac{d}{dt}\left(\phi_{i,j}\right) \approx Residual_{i,j} \qquad i = 1,2,\ldots,NR \quad ; \quad j = 1,2,\ldots,NZ \tag{4.14}$$

where $Residual_{i,j}$ is the finite difference representation of the spatial derivatives. The

system of ODEs together with the initial and boundary conditions are integrated in time by means of any of the explicit or implicit ODE integration methods such as Euler's method, ROW methods, Backward Differentiation Formula (BDF), Adams-Moulton etc.

As mentioned earlier, the most important feature of the MOL approach is that not only does it have the simplicity of the explicit methods but also the superiority of the implicit ones as higher-order implicit time integration methods are employed in the solution of the resulting system of ODEs. There exist many efficient and reliable stiff ODE solvers in the open literature. However, it is very important to select a suitable solver considering the type and dimension of the physical system, desired level of accuracy and execution time. In consideration of this, the code developed in this study is equipped with two different ODE solvers which will be briefly described below:

ROWMAP: ROWMAP is based on the ROW-methods of order 4 and uses Krylov tecniques for the solution of linear systems [67]. By a special multiple Arnoldi process, the order of the basic method is preserved with small Krylov dimensions. Step size control is done by sixth order method embedded in the code. The source code of ROWMAP can be obtaibed from the netsite, *http://www.matematik.uni-halle.de/institute/numerik/software*.

LSODES: LSODES solves stiff and non-stiff systems of the form $dy/dt = f$. Non-stiff systems are handled by Adams methods (predictor-corrector) whereas BDF (GEAR methods) are used for the stiff cases. It determines the sparsity structure on its own (or optionally accepts this information from the user) and then uses parts of the Yale Sparse Matrix Package (YSMP) to solve the linear systems that arise, by a sparse (direct) LU factorization/backsolve method. LSODES supersedes, and improves upon, the older and well known GEARS package. The source code of LSODES and its dependencies can be obtained from the netsite *http://www.netlib.org/odepack/* and detailed description on the solver can be found elsewhere [68].

Based on the previous experience with a MOL based CFD code for the simulation of chemically reacting flows, integration of ODEs is performed by using LSODES [68]

in the present investigation.

4.5 Numerical Solution of the Pressure Equation

It has been shown that the present algorithm necessitates the solution of a elliptic pressure equation (also known as the pressure Poisson equation) at each momentum corrector stage. Utilization of classical iterative methods for the solution of pressure Poisson equation in time-dependent computations is compuationally expensive and takes most of the computing effort. Direct or multigrid methods on the other hand are more attractive for these type of calculations owing to their efficiency and robustness [69]. Under the light of these facts, the present algorithm was equipped with a hybrid multigrid/direct solver namely MUDPACK [70]. MUDPACK is a collection of FORTRAN subprograms for the solution elliptic PDEs on any bounded rectangular domain in two-dimensions (not restricted to cartesian coordinates) and boxes in three-dimensions with any combination of boundary conditions. The multigrid iteration used in MUDPACK requires far less storage and computation than direct methods for non-separable PDEs and is competitive with cyclic reduction for separable PDEs. Moreover, the second-order accurate results produced by the solver can be improved to fourth-order accuracy with ease using the method of deffered corrections. The package and detailed documentation on the solvers can be obtained from [71].

For the solution of elliptic pressure equation of the form

$$\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^{*} = Residual_p^{*} \tag{4.15}$$

where

$$\begin{aligned} Residual_p^{*} \approx & - \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}\left(\rho^{*}v^{**}\right)+\frac{\rho^{*}v^{**}}{r}+\frac{\partial}{\partial z}\left(\rho^{*}u^{**}\right)\right) \\ & + \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}\left(\rho^{n}v^{*}\right)+\frac{\rho^{n}v^{*}}{r}+\frac{\partial}{\partial z}\left(\rho^{n}u^{*}\right)\right) \\ & - LHS^{*}+RHS^{*}, \end{aligned} \tag{4.16}$$

$$\begin{aligned} LHS = & \frac{\partial^2}{\partial z^2}\left(\rho^{n}u^{*}u^{*}\right)+\frac{\partial^2}{\partial r^2}\left(\rho^{n}v^{*}v^{*}\right)+\frac{2}{r}\frac{\partial}{\partial r}\left(\rho^{n}u^{*}v^{*}\right) \\ & + \frac{2}{r}\frac{\partial}{\partial z}\left(\rho^{n}u^{*}v^{*}\right)+2\frac{\partial^2}{\partial r\partial z}\left(\rho^{n}u^{*}v^{*}\right), \end{aligned} \tag{4.17}$$

$$RHS = -\left(\frac{\partial^2\tau_{zz}}{\partial z^2}+\frac{\partial^2\tau_{rr}}{\partial r^2}+2\frac{\partial^2\tau_{rz}}{\partial r\partial z}+\frac{2}{r}\frac{\partial\tau_{rr}}{\partial r}+\frac{2}{r}\frac{\partial\tau_{rz}}{\partial z}-\frac{1}{r}\frac{\partial\tau_{\theta\theta}}{\partial r}\right)^*, \qquad (4.18)$$

MUDPACK requires that the coefficients of the elliptic equation and value of the $Residual_p^*$ term at each grid point are specified. Due to the presence of time-advanced terms in the right-hand-side, Equation (4.16) is implicit in time and hence cannot be directly used for the calculation of $Residual_p^*$ term. The generic procedure to surmount this problem was outlined in Chapter 3 and it was shown that after necessary manipulations, the explicit pressure equation was obtained as follows

$$\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* - \frac{1}{\Delta t^2}\frac{p^*}{RT^*} = Residual_p^* \qquad (4.19)$$

where

$$\begin{aligned} Residual_p^* &= -\frac{\rho^n}{\Delta t^2}+\frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*)+\frac{\rho^n v^*}{r}+\frac{\partial}{\partial z}(\rho^n u^*)\right) \qquad (4.20)\\ &- LHS^*+RHS^*. \end{aligned}$$

Inspection of Equation (4.19) reveals that it is variable coefficient Poisson equation (due to the temperature term on the left-hand-side) solution of which is subject to stability problems [72]. Therefore, instead of solving Equation (4.19) for the determination of pressure field, a different approach is adopted in the present study: Recall that substitution of equation of continuity (Equation (3.12)) into Equation (4.15) gives

$$\begin{aligned} \left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* &= \frac{1}{\Delta t}\left[\frac{\rho^*-\rho^n}{\Delta t}\right] \qquad (4.21)\\ &+ \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*)+\frac{\rho^n v^*}{r}+\frac{\partial}{\partial z}(\rho^n u^*)\right)\\ &- LHS^*+RHS^*. \end{aligned}$$

Considering that the density variation in the low Mach number flows of interest is mostly due to temperature and concentration rather than pressure [19] and the former two remain unchanged during a momentum corrector stage, the first term in brackets in Equation (4.21) is dropped yielding the explicit constant coefficient Poisson equation for pressure as follows

$$\left(\frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{\partial^2 p}{\partial z^2}\right)^* = Residual_p^* \qquad (4.22)$$

where in this case $Residual_p^*$ term is defined as

$$Residual_p^* = \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*) + \frac{\rho^n v^*}{r} + \frac{\partial}{\partial z}(\rho^n u^*)\right) - LHS^* + RHS^*. \qquad (4.23)$$

The merits of using Equation (4.22) instead of Equation (4.19) are not only that the former is more stable in terms of numerical solution but also that the computational power requirement for its solution is much less than that required for the latter.

In an attempt to verify the validity of the approach used, the right-hand-side of Equation (4.21) was decomposed into two terms;

$$term1 = \frac{1}{\Delta t}\left[\frac{\rho^* - \rho^n}{\Delta t}\right], \qquad (4.24)$$

$$term2 = \frac{1}{\Delta t}\left(\frac{\partial}{\partial r}(\rho^n v^*) + \frac{\rho^n v^*}{r} + \frac{\partial}{\partial z}(\rho^n u^*)\right) - LHS^* + RHS^*, \qquad (4.25)$$

and the magnitudes of $term1$ and $term2$ were compared. The comparison revealed that $term2$ is three orders of magnitude greater than the omitted $term1$ which fortifies the utilization of Equation (4.23) for the computation of $Residual_p^*$ term.

4.5.1 Specification of Boundary Conditions for Pressure Equation

Specification of boundary conditions is flexible in MUDPACK; use of any combination of periodic, Dirichlet and mixed-derivative boundary conditions is possible. For a typical solution of pressure equation, at least one Drichlet type boundary condition is specified as an inlet or outlet condition. Once this is set, the rest of the boundary conditions can be obtained using momentum equations. Table 4.1 summarizes the boundary conditions utilized in the present study.

Table 4.1: Boundary conditions for pressure equation

BC1:	@ the center,	$\forall z \Lambda \forall t$:	$\frac{\partial p}{\partial r} = 0$
BC2:	@ the wall,	$\forall z \Lambda \forall t$:	$\frac{\partial p}{\partial r}$ specified using r-momentum equation
BC3:	@ the inlet,	$\forall r \Lambda \forall t$:	$\frac{\partial p}{\partial z}$ specified using z-momentum equation
BC4:	@ the outlet,	$\forall r \Lambda \forall t$:	$p = p_{ref}$

As can be seen, the only Dirichlet type boundary condition was imposed at the outlet by setting the pressure at the outlet to its reference value ($p = p_{ref}$). At the centerline, symmetry condition was utilized. The wall and inlet conditions are derived from *r*- and *z*-momentum equations, respectively by making necessary simplifications as follows

$$\frac{\partial p}{\partial r} = -\frac{1}{r}\frac{\partial}{\partial r}(r\rho vv) - \left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rr}) + \frac{\partial \tau_{zr}}{\partial z} - \frac{1}{r}\tau_{\theta\theta}\right) + \rho g_r, \tag{4.26}$$

$$\frac{\partial p}{\partial z} = -\frac{\partial}{\partial z}(\rho uu) - \left(\frac{1}{r}\frac{\partial}{\partial r}(r\tau_{rz}) + \frac{\partial \tau_{zz}}{\partial z}\right) + \rho g_z. \tag{4.27}$$

4.6 Computation of Radiative Source Term

For the computation of radiative source term, use has been made of previously developed in-house radiation codes based on the MOL solution of Discrete Ordinates Method (DOM) for gray [49] and non-gray [50] media by means of coupling them to the CFD code developed in this study. In the next sections, description of DOM and its MOL solution for gray media followed by the extension of the scheme to non-gray media will be presented.

4.6.1 Discrete Ordinates Method

DOM is based on representation of continuous angular domain by a discrete set of ordinates with appropriate angular weights, spanning the total solid angle of 4π steradians. The discrete ordinates representation of the RTE (Equation (2.79)) for an absorbing-emitting gray medium in axisymmetric cylindrical coordinate system takes the following form

$$\frac{\mu_m}{r}\frac{\partial}{\partial r}(rI^m) - \frac{1}{r}\frac{\partial}{\partial \phi}(\eta_m I^m) + \xi_m \frac{\partial I^m}{\partial z} = -\kappa I^m + \kappa I_b \tag{4.28}$$

where $I^m[\equiv I(r,z;\theta,\phi)]$ is the total radiation intensity at position (r,z) in the discrete direction Ω_m. The terms on the left hand-side represents the gradient of intensity in curvilinear coordinates, and the two terms on the right hand-side stand for the changes in intensity due to absorption and emission, respectively.

The angular derivative term is discretized by introducing an angular redistribution term $\gamma_{m,\ell\pm1/2}$ proposed by Carlson and Lathrop [73] which can be represented by the following expression

$$\left[\frac{\partial}{\partial\phi}(\eta_m I^m)\right]_{\Omega_m=\Omega_{m,\ell}} = \left(\frac{\gamma_{m,\ell+1/2}I^{m,\ell+1/2} - \gamma_{m,\ell-1/2}I^{m,\ell-1/2}}{w_{m,\ell}}\right) \tag{4.29}$$

where $I^{m,\ell+1/2}$ and $I^{m,\ell-1/2}$ are radiation intensities in directions $m,\ell+1/2$ and $m,\ell-1/2$, respectively which define the edges of angular range of $w_{m,\ell}$ and can be formulated as

$$I^{m,\ell+1/2} = \frac{I^{m,\ell} + I^{m,\ell+1}}{2}, \tag{4.30}$$

and

$$I^{m,\ell-1/2} = \frac{I^{m,\ell-1} + I^{m,\ell}}{2}. \tag{4.31}$$

The angular redistribution term can be expressed as

$$\gamma_{m,\ell+1/2} = \gamma_{m,\ell-1/2} + \mu_{m,\ell}w_{m,\ell} \qquad\qquad \ell = 1,2,\ldots,L \tag{4.32}$$

where, L is the maximum value of ℓ for a particular m which depends on the number of discrete directions used to represent one octant of a sphere dicatated by the angular quadrature scheme (S_N). The quadrature ordinates and weights for S_N approximations are given in Appendix A. Further details of angular redistribution term and estimation of the angular derivative can be found in [49, 74]. Substitution of Equation (4.29) into (4.28) yields the final form of the discrete ordinates equation for axisymmetric cylindrical geometry

$$\begin{aligned}\frac{\mu_{m,\ell}}{r}\frac{\partial}{\partial r}\left(rI^{m,\ell}\right) \quad &- \quad \frac{1}{r}\frac{\left(\gamma_{m,\ell+1/2}I^{m,\ell+1/2} - \gamma_{m,\ell-1/2}I^{m,\ell-1/2}\right)}{w_{m,\ell}} \\ &+ \quad \xi_{m,\ell}\frac{\partial I^{m,\ell}}{\partial z} = -\kappa I^{m,\ell} + \kappa I_b.\end{aligned} \tag{4.33}$$

For diffuse surfaces, boundary conditions required for the solution of Equation (4.33) on the surface of the enclosure can be written as

at z=0;

$$I^{m,\ell} = \varepsilon_w I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi}\sum_{m',\ell'} w_{m',\ell'}\left|\xi_{m',\ell'}\right| I^{m',\ell'} \qquad\qquad \xi_{m,\ell} > 0 \tag{4.34}$$

at z=L;

$$I^{m,\ell} = \varepsilon_w I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi} \sum_{m',\ell'} w_{m',\ell'} \xi_{m',\ell'} I^{m',\ell'} \qquad \xi_{m,\ell} < 0 \qquad (4.35)$$

at r=R;

$$I^{m,\ell} = \varepsilon_w I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi} \sum_{m',\ell'} w_{m',\ell'} \mu_{m',\ell'} I^{m',\ell'} \qquad \mu_{m,\ell} < 0 \qquad (4.36)$$

at r=0;

$$I^{m,\ell} = I^{m',\ell'} \qquad \mu_{m,\ell} > 0 \qquad (4.37)$$

where m,ℓ and m',ℓ' denote outgoing and incoming directions respectively. In the present study, the wall surface, and inlet and outlet imaginary surfaces were assumed to be radiatively black.

4.6.2 MOL Solution of DOM for Gray Media

The solution of discrete ordinates equations with MOL is carried out by adoption of the false-transients approach which involves incorporation of a pseudo-time derivative of intensity into the discrete ordinates equations [75]. Adoption of the false-transient approach to Equation 4.33 yields

$$\begin{aligned} k_t \frac{\partial I^{m,\ell}}{\partial t} = & - \frac{\mu_{m,\ell}}{r} \frac{\partial (r I^{m,\ell})}{\partial r} + \frac{1}{r} \frac{(\gamma_{m,\ell+1/2} I^{m,\ell+1/2} - \gamma_{m,\ell-1/2} I^{m,\ell-1/2})}{w_{m,\ell}} \\ & - \xi_{m,\ell} \frac{\partial I^{m,\ell}}{\partial z} - \kappa I^{m,\ell} + \kappa I_b \end{aligned} \qquad (4.38)$$

where t is the pseudo-time variable and k_t is a time constant with dimension $[(m/s)^{-1}]$ which is introduced to maintain dimensional consistency in the equation and it is taken as unity.

Following the MOL approach, the system of PDEs together with the initial and boundary conditions are then transformed into an ODE initial value problem. The transformation is carried out by representing the spatial derivatives with the algebraic finite-difference approximations details of which can be found in [49]. Time integration of the resulting system of ODEs is carried out in a similar fashion as

in the CFD code. Starting from an initial condition for radiation intensities in all directions, the system of ODEs is integrated by means of a powerful ODE solver until steady state. Any initial condition can be chosen to start the integration, as its effect on the steady state solution decays to insignificance. To stop the integration at steady state, a convergence criterion based on the difference of the intensities at consecutive time-steps is utilized at all grid points.

Once the intensity distribution is determined by the solution of Equation (4.38) together with its boundary conditions, the radiative energy source term can be computed by

$$\nabla \cdot \mathbf{q_R} = \kappa \left(4\pi I_b - \sum_m \sum_\ell w_{m,\ell} I^{m,\ell} \right) \tag{4.39}$$

In the present study, the spatial derivatives appearing in the RTE are discretized using two-point upwind differencing scheme. The S_4 order of approximation was found to be optimum by succesive refinement studies [62]. The temperature and concentration dependent absorption coefficients for the gray gas were calculated using Leckner's correlations [76]. RKF45 (Runge-Kutta-Fehlberg integration) subroutine [38], which is an adaptive, fourth order accurate ODE solver, was utilized for time integration.

4.6.3 MOL Solution of DOM for Non-Gray Media

In order to determine the radiative intensity distribution for non-gray media, the whole spectrum is discretized into wave number intervals within which the radiative properties are assumed to be constant. All wave number intervals having absorption coefficients within a certain range are combined into a gray gas which necessitates the modification of RTE (Equation (2.79)) as follows [77, 78]

$$\frac{dI_j}{ds} = \frac{\mu}{r}\frac{\partial}{\partial r}\left(rI_j\right) - \frac{1}{r}\frac{\partial}{\partial \phi}\left(\eta I_j\right) + \xi \frac{dI_j}{dz} = -\kappa_j (a_j I_b - I_j) \tag{4.40}$$

where the subscript j denotes the spectral division and a_j is the blackbody weights determined from the standard blackbody distribution function of the Wide Band Correlated-k model (WBCK) [79] together with the absorption-line distribution functions of the Spectral Line-Based Weighted Sum of Gray Gases

model (SLW) [80]. $I_b(\equiv \sigma T^4/\pi)$ is the blackbody radiation intensity at the surface temperature. Details regarding to the models and estimation of spectral properties can be found elsewhere [50, 60].

Similar to the procedure described previously, application of DOM and false transients approach to Equation (4.40) yields

$$k_{t^*}\frac{\partial I_j^{m,\ell}}{\partial t} = - \frac{\mu_{m,\ell}}{r}\frac{\partial (rI_j^{m,\ell})}{\partial r} + \frac{1}{r}\frac{(\gamma_{m,\ell+1/2}I_j^{m,\ell+1/2} - \gamma_{m,\ell-1/2}I_j^{m,\ell-1/2})}{w_{m,\ell}} - \xi_{m,\ell}\frac{\partial I_j^{m,\ell}}{\partial z} + \kappa_j(a_j I_b - I_j^{m,\ell}). \quad (4.41)$$

Equation (4.41) togheter with the following boundary conditions

at z=0;

$$I_j^{m,\ell} = \varepsilon_w a_j I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi}\sum_{m',\ell'} w_{m',\ell'}\left|\xi_{m',\ell'}\right| I_j^{m',\ell'} \qquad \xi_{m,\ell} > 0 \quad (4.42)$$

at z=L;

$$I_j^{m,\ell} = \varepsilon_w a_j I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi}\sum_{m',\ell'} w_{m',\ell'}\xi_{m',\ell'} I_j^{m',\ell'} \qquad \xi_{m,\ell} < 0 \quad (4.43)$$

at r=R;

$$I_j^{m,\ell} = \varepsilon_w a_j I_b(\mathbf{r_w}) + \frac{(1-\varepsilon_w)}{\pi}\sum_{m',\ell'} w_{m',\ell'}\mu_{m',\ell'} I_j^{m',\ell'} \qquad \mu_{m,\ell} < 0 \quad (4.44)$$

at r=0;

$$I_j^{m,\ell} = I_j^{m',\ell'} \qquad \mu_{m,\ell} > 0 \quad (4.45)$$

are solved for each gray gas using the same procedure described in Section 4.6.2 to give the spectral intensity distribution for the entire spectrum. Once the intensity disributions are known, radiative energy source term can be evaluated using

$$\nabla \cdot \mathbf{q_R} = \sum_j^{NG} \kappa_j \left(4\pi a_j I_b - \sum_m \sum_\ell w_{m,\ell} I_j^{m,\ell}\right) \quad (4.46)$$

where NG is the number of gray gases the spectrum is divided into which is taken as 10 for both CO_2 and H_2O in the present study.

4.6.4 Coupling Strategy

The coupling strategy is mainly based on regular transfer of temperature and concentrations of participating gases (CO_2 and H_2O) obtained by the CFD code to the radiation code which in turn provides source term field as the solution propogates in time [2, 62]. Owing to the nature of radiation transport, radiative heat transfer computations can be performed on much coarser grid resolutions when compared to that required for CFD. Hence rather than using identical grid resolutions for both CFD and radiation codes, two different resolutions were utilized: a fine mesh for CFD code and a coarse one enabling economic computation of radiative source term (Figure (4.1)) both of which are uniformly distributed.

The computations commence with the CFD code making temperature and species distributions available for the radiation code. Temperature and concentrations at the overlapping grid points of the coarse and fine meshes are transferred to the radiation code which calculates radiative source term for the CFD code. Source terms on the coarse mesh are then redistributed to fine CFD mesh via 2-D linear interpolation. This cyclic loop is continued until steady-state which is dictated by the CFD code.

CHAPTER 5

STRUCTURE AND OPERATION OF THE COMPUTER CODE

5.1 Preamble

The purpose of this chapter is to give the necessary information on the structure and organization of the code developed in the present study for the simulation of reacting radiating flows. Also presented in this chapter is the mode of operation and the range of applicability of the code which all together will serve as a manual for modification and testing purposes.

5.2 Range of Applicability

The computer code developed in this study is a general purpose program for the simultaneous solution of continuity, momentum, energy species equations along with a Poisson-type pressure equation for the computation of unsteady reacting radiating flows. User can select one of the seven finite-rate reaction mechanims with varying complexity and incorporation of different mechanisms on top of the existing roster is possible. Both gray and spectral radiation modules are available for radiative heat transfer calculations. Owing to the modular nature of the code, simulation of isothermal/non-isothermal non-reacting flows can also be achieved by minor modifications in the code. Due to the fact that the it is a teaching-oriented in-house code rather than a *black-box* commercial one, the user must have adequate knowledge and experience on the principles of CFD, combustion and radiation, before attempting to use or modify it. Nevertheless, the program is written in a relatively straightforward fashion which is readily amenable to modification. Most subroutines are problem-independent and in many cases only minor modifications

are required for the adaptation of the problem-dependent subroutines to conditions other than covered in the present study.

5.3 Structure of the Code

The structure of the computer program is illustrated in Figure (5.1). As can be seen, the code has been organized in a modular fashion as much as possible with separate subprograms having different tasks. The main program is composed of two major subroutines, INITIAL and TIME_INTEGRATION both of which having large number of dependencies. In accordance with the numerical solution algorithm (splitting of operations), TIME_INTEGRATION subroutine has been divided into individual routines for the integration of momentum, energy and species equations (MOMENTUM_INTEGRATOR, ENERGY_INTEGRATOR and SPECIES_INTEGRATOR). This way, application of the code to a variety of flow scenarios can be performed with the desired combination of predictor-corrector sequence and without modifying its core structure. For instance, user can bypass the SPECIES_INTEGRATOR for the simulation of non-reacting flows by simply setting the number of predictor and corrector stages for the solution of species equations to zero. Moreover, a different ODE solver together with a different time step can be employed for each integrator, a feature which might reduce the CPU time significantly for problems with stiff chemistry.

5.4 Mode of Operation

The detailed algorithm of the computer program developed for the simulation of reacting radiating flows is presented in Figures (5.2) to (5.22). As explained earlier, solution of conservation equations is based on a sequence of predictor-corrector stages which involves the evaluation of time derivative vectors (system of ODEs) and advancing in time by means of integration of the ODEs.

Initially, all dependent variables are known *a priori* at the begining of each cycle, either as a result of the previous cycle or from the prescribed initial conditions. The integration sequence commences with the predictor stage and followed by the desired

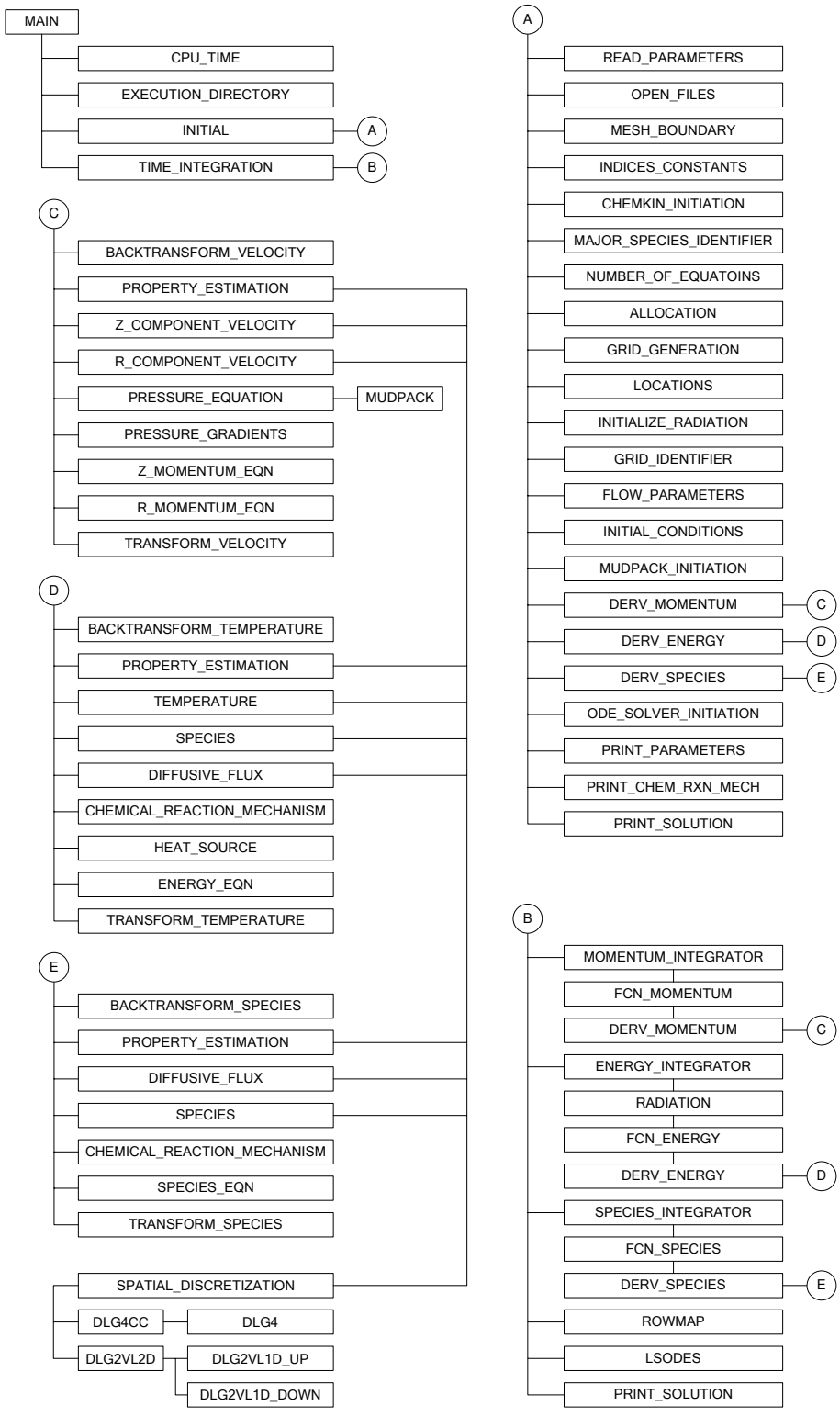


Figure 5.1: Organization of the computer code.

number of corrector stages. Time derivatives of the conservation equations for each stage are calculated by the evaluation of the physical and thermodynamic properties and spatial derivatives based on the values of the present cycle. Radiative heat source term appearing in the energy equation is calculated using the radiation module at user defined time steps (Δt). A pressure equation is solved for the determination of pressure field at momentum corrector stages. First, second and third intermediate fields are obtained as a result of integration of the time derivatives in the predictor, first and second corrector stages, respectively. At the end of final corrector stage, the advanced time level velocity, pressure, temperature and species fields are obtained concluding the algorithm for one time step (Δt). This cyclic procedure is repeated until steady-state is reached.

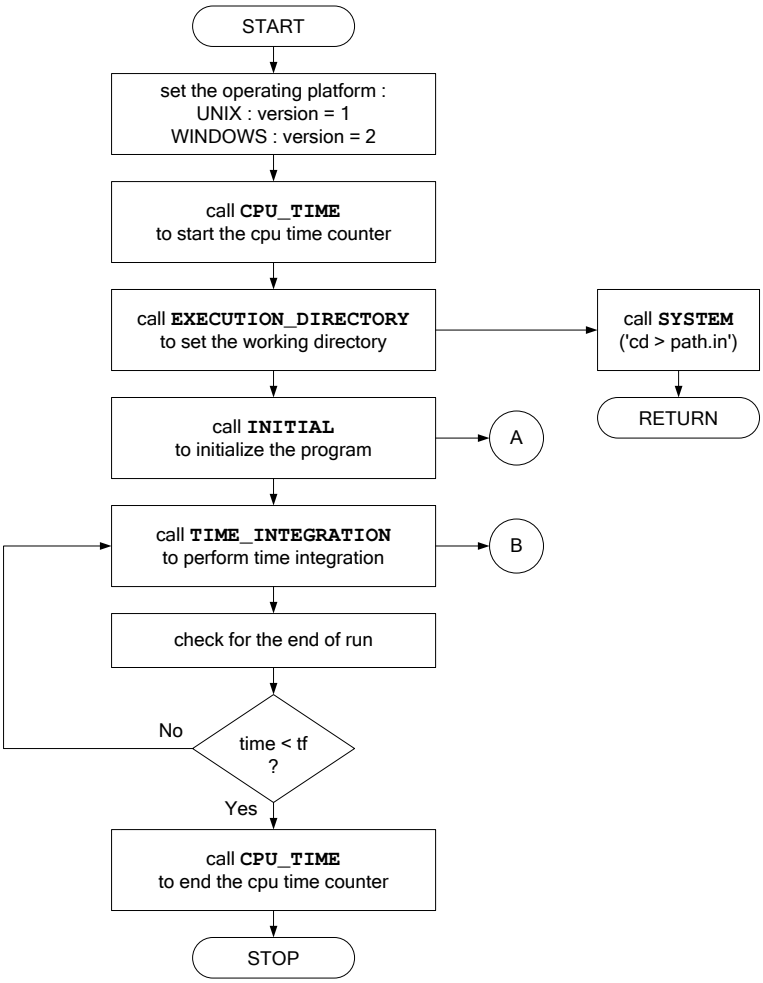


Figure 5.2: Algorithm of the main program.

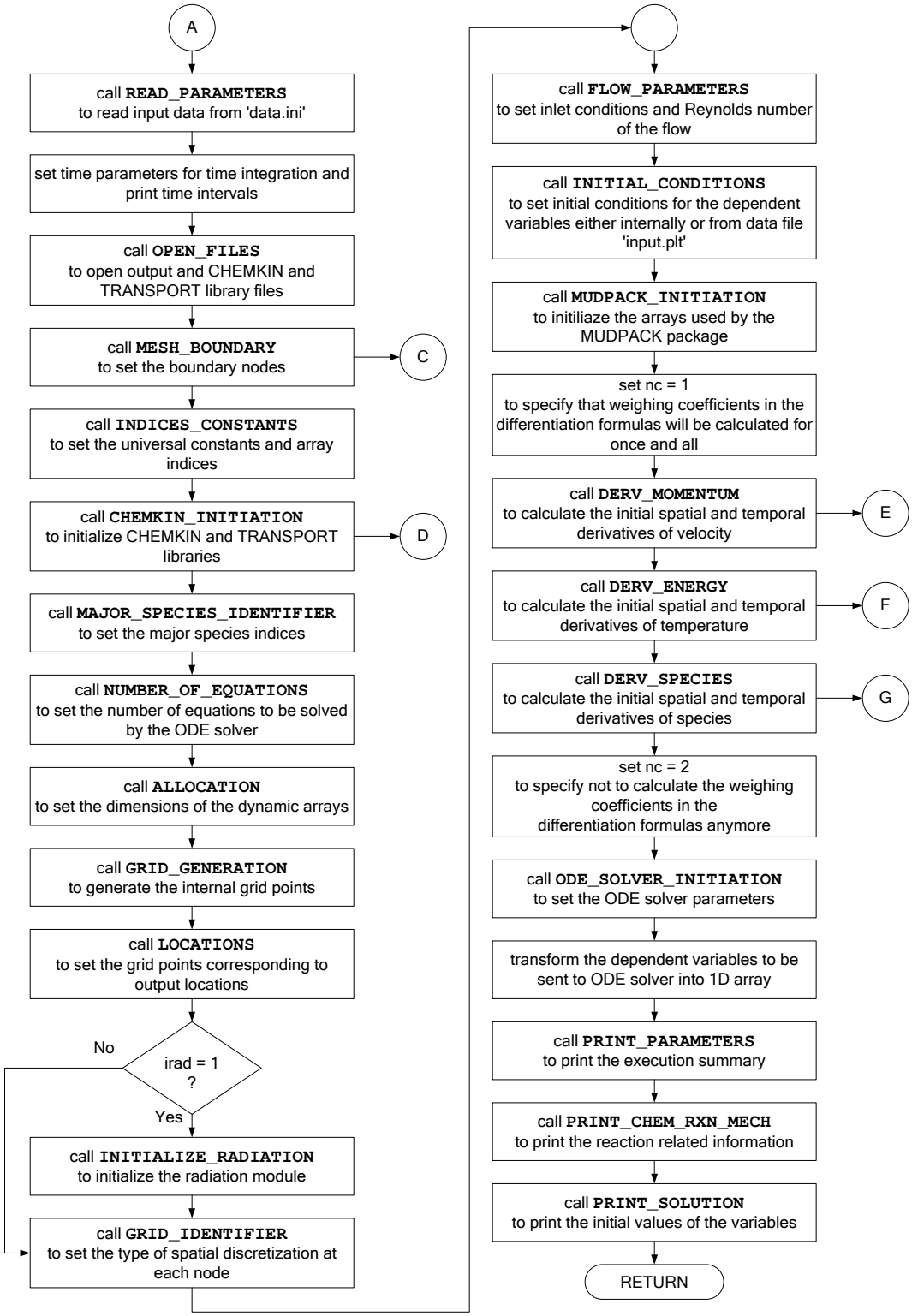


Figure 5.3: Algorithm of subroutine INITIAL.

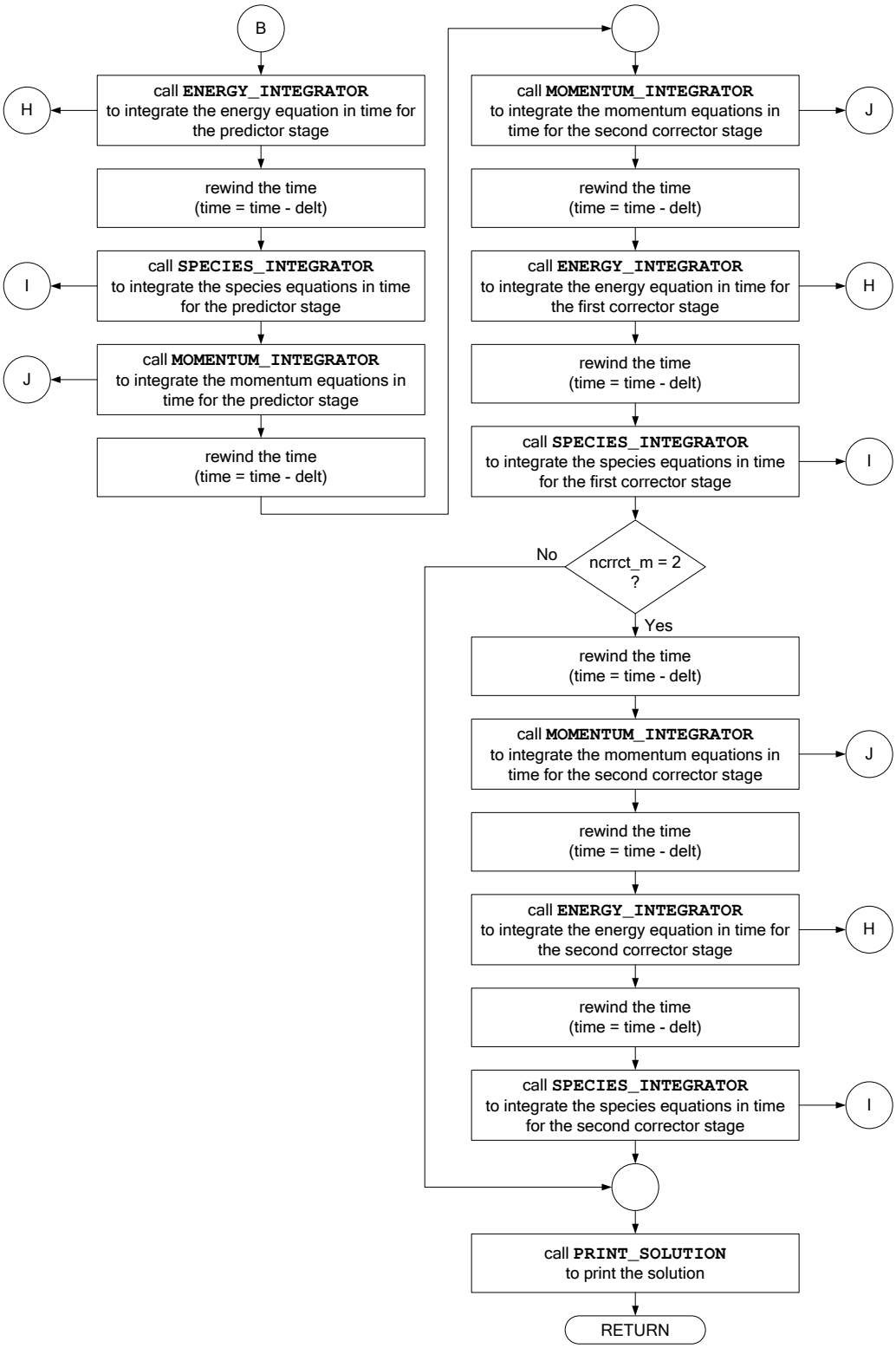


Figure 5.4: Algorithm of subroutine TIME_INTEGRATION.

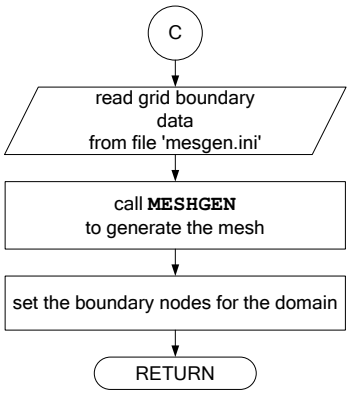


Figure 5.5: Algorithm of subroutine MESH_BOUNDARY.

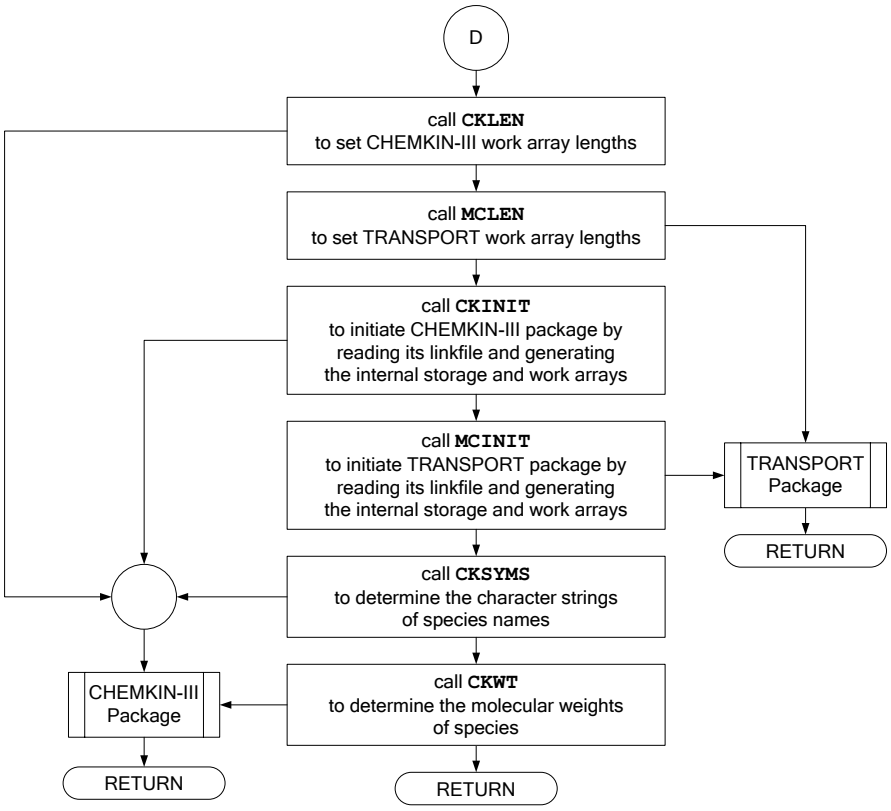


Figure 5.6: Algorithm of subroutine CHEMKIN_INITIATION.

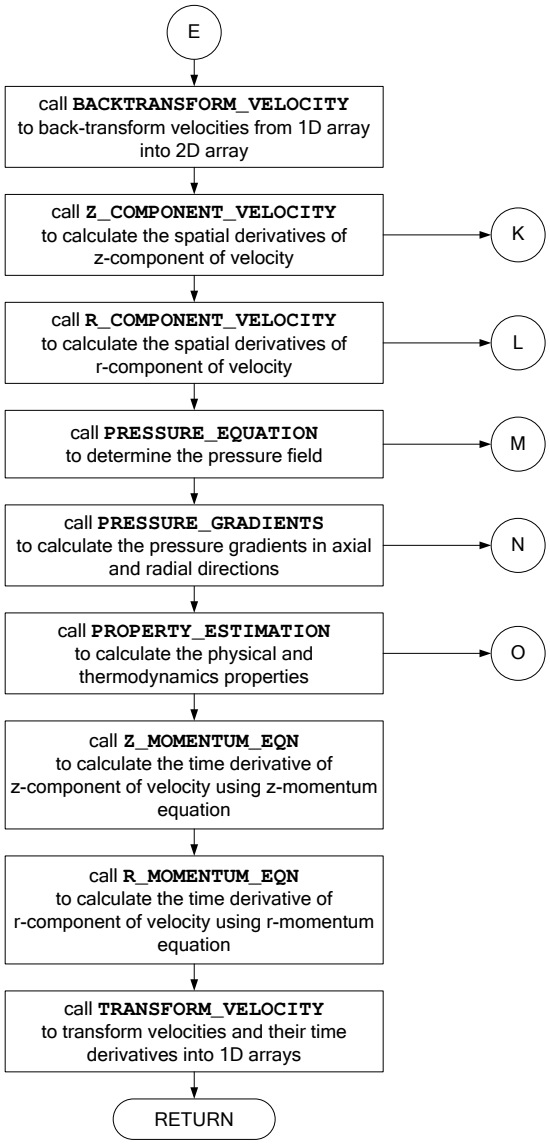


Figure 5.7: Algorithm of subroutine DERV_MOMENTUM.

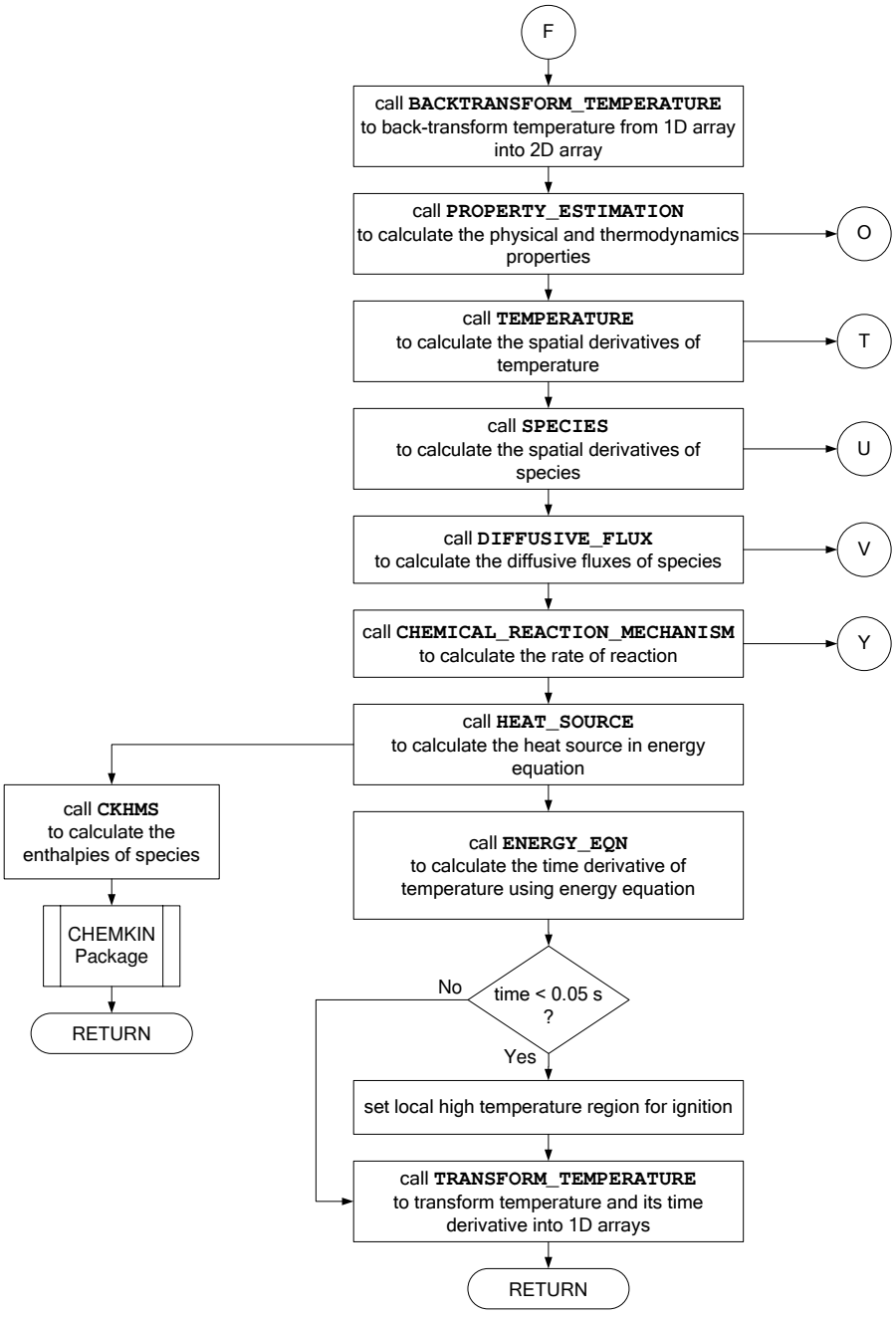


Figure 5.8: Algorithm of subroutine DERV_ENERGY.

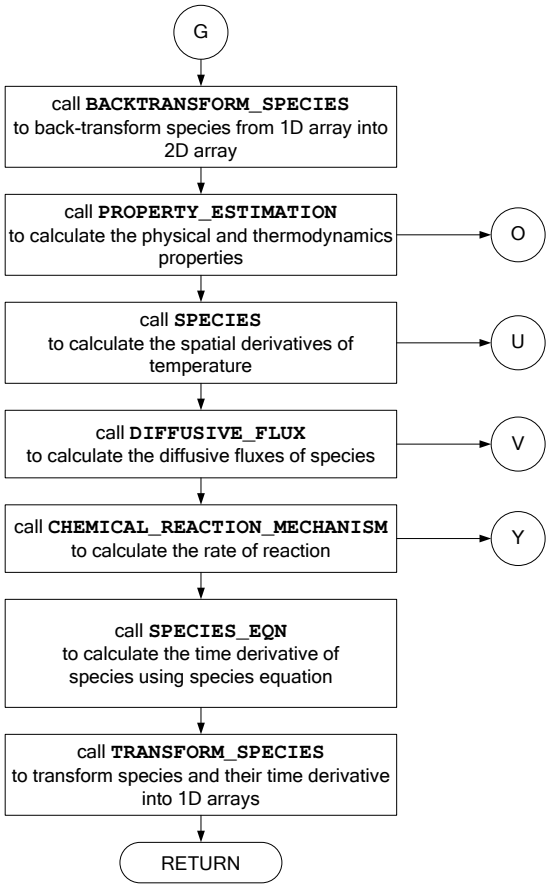


Figure 5.9: Algorithm of subroutine DERV_SPECIES.

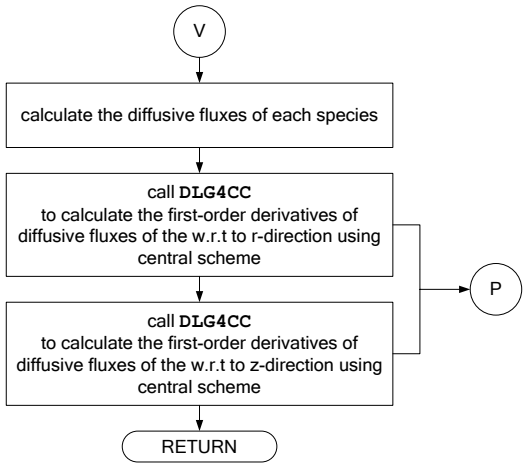


Figure 5.10: Algorithm of subroutine DIFFUSIVE_FLUX.

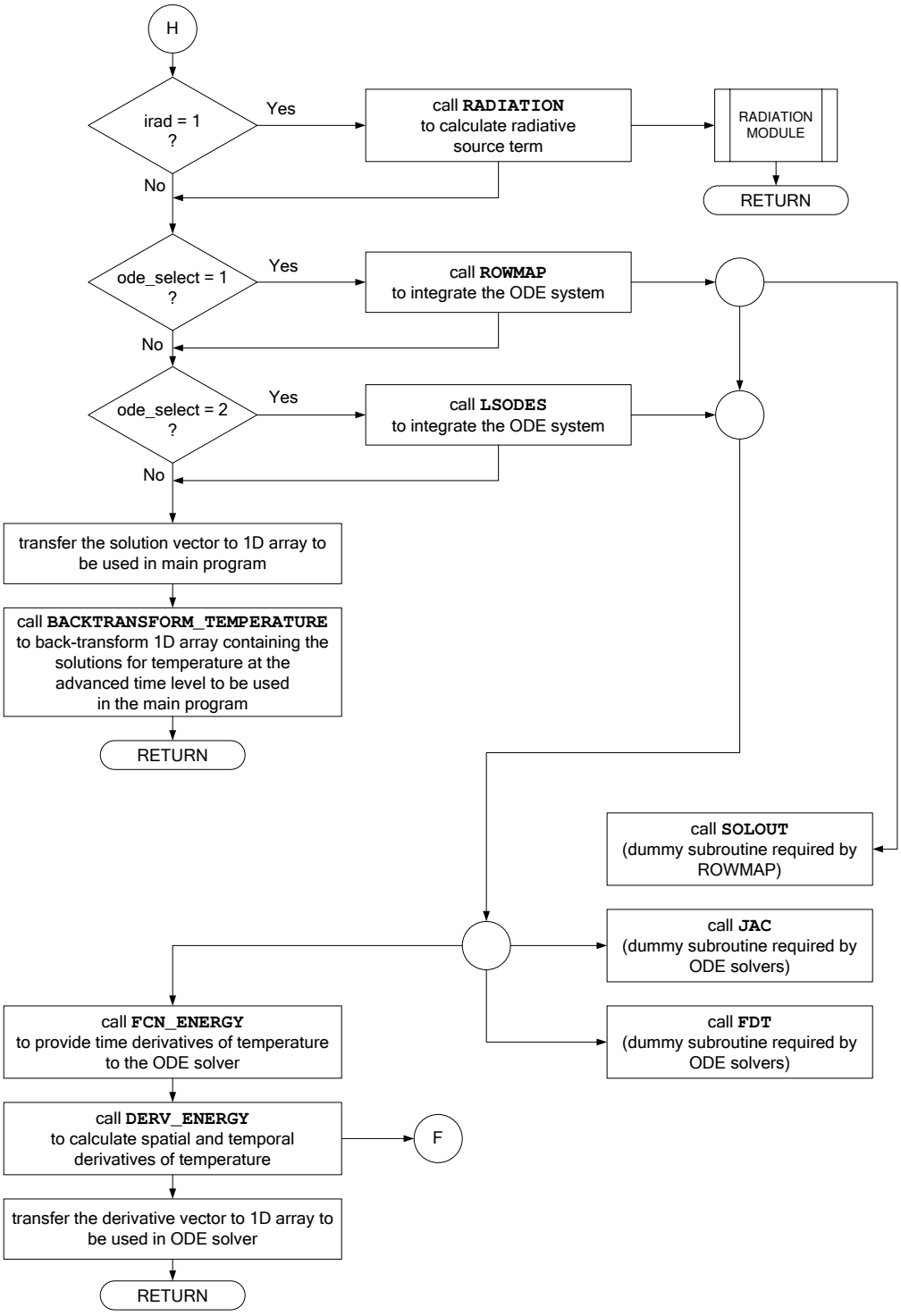


Figure 5.11: Algorithm of subroutine ENERGY_INTEGRATOR.

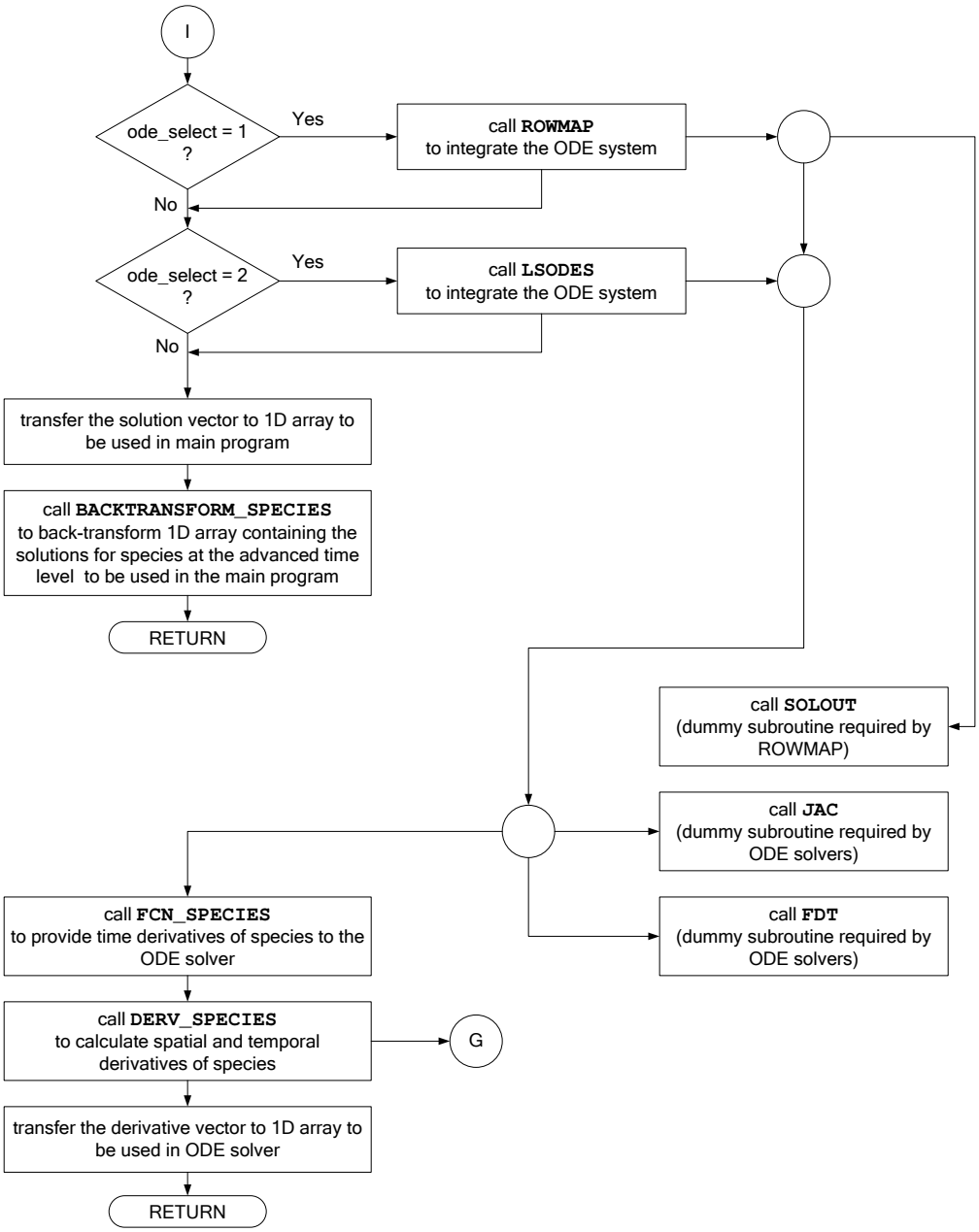


Figure 5.12: Algorithm of subroutine SPECIES_INTEGRATOR.

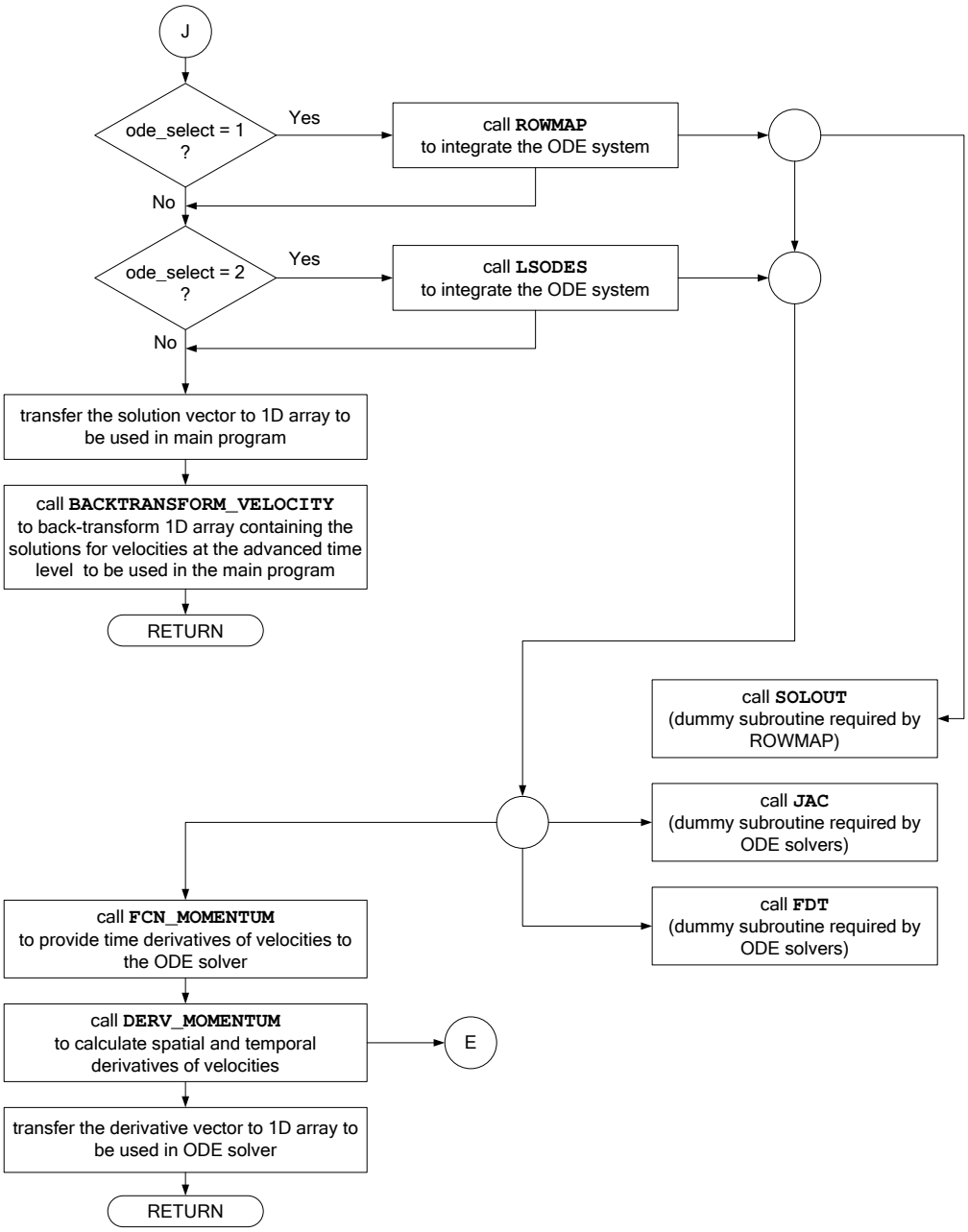


Figure 5.13: Algorithm of subroutine MOMENTUM_INTEGRATOR.

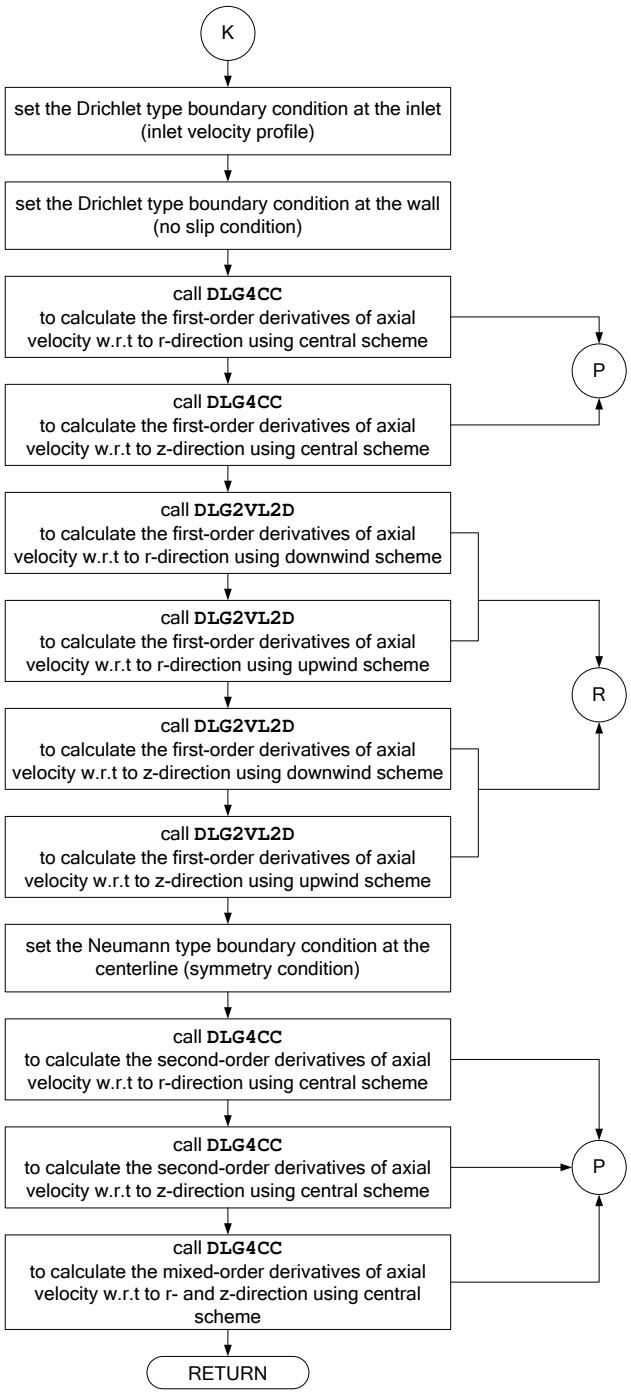


Figure 5.14: Algorithm of subroutine Z_COMPONENT_VELOCITY.

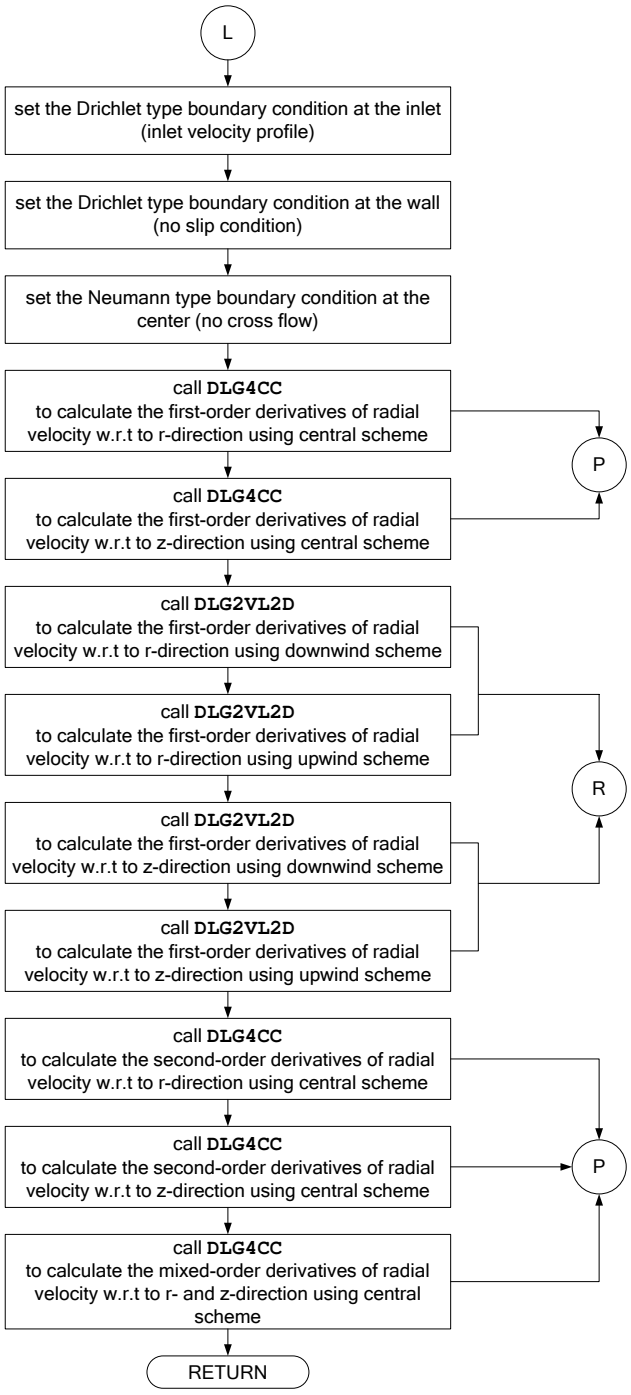


Figure 5.15: Algorithm of subroutine R_COMPONENT_VELOCITY.

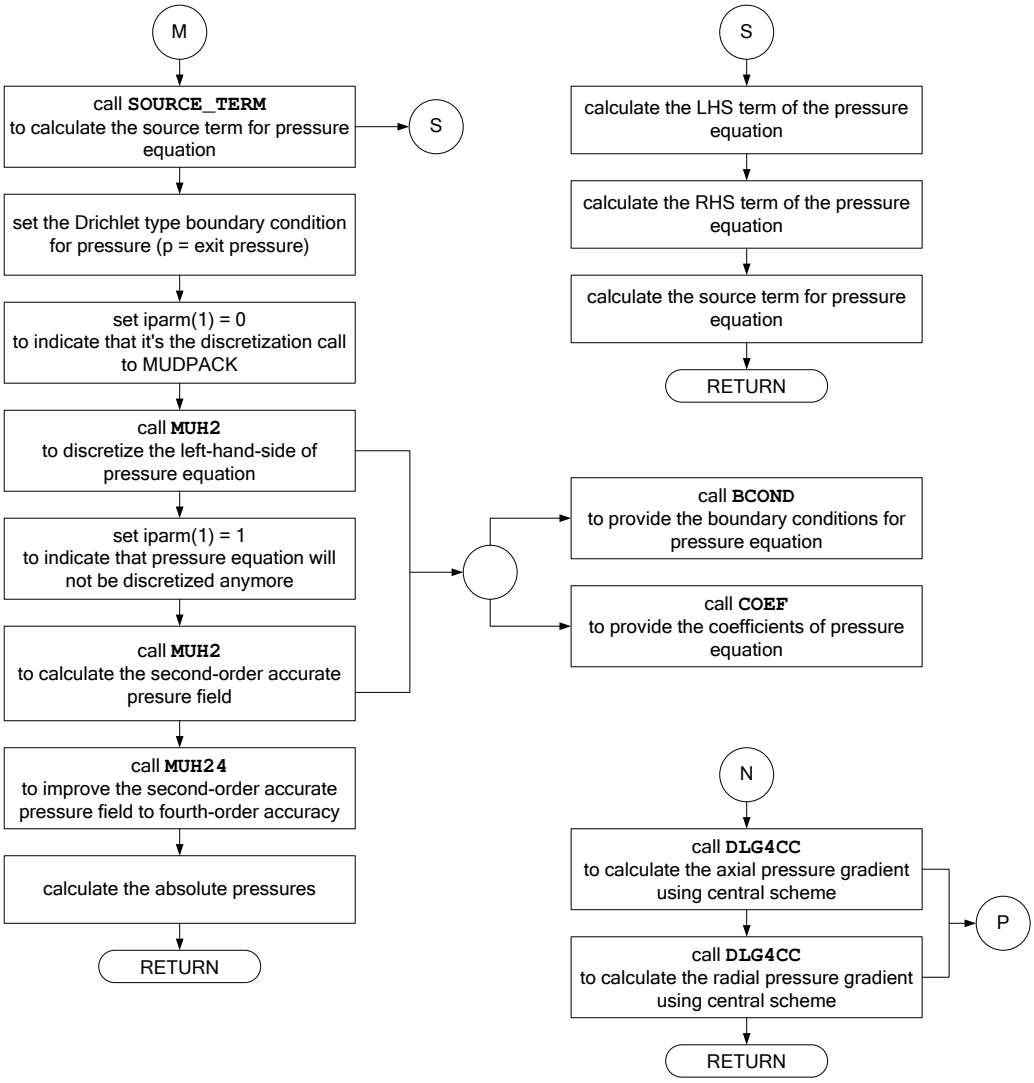


Figure 5.16: Algorithm of subroutines SOURCE_COMPUTATION, PRESSURE_EQUATION, PRESSURE_GRADIENTS.

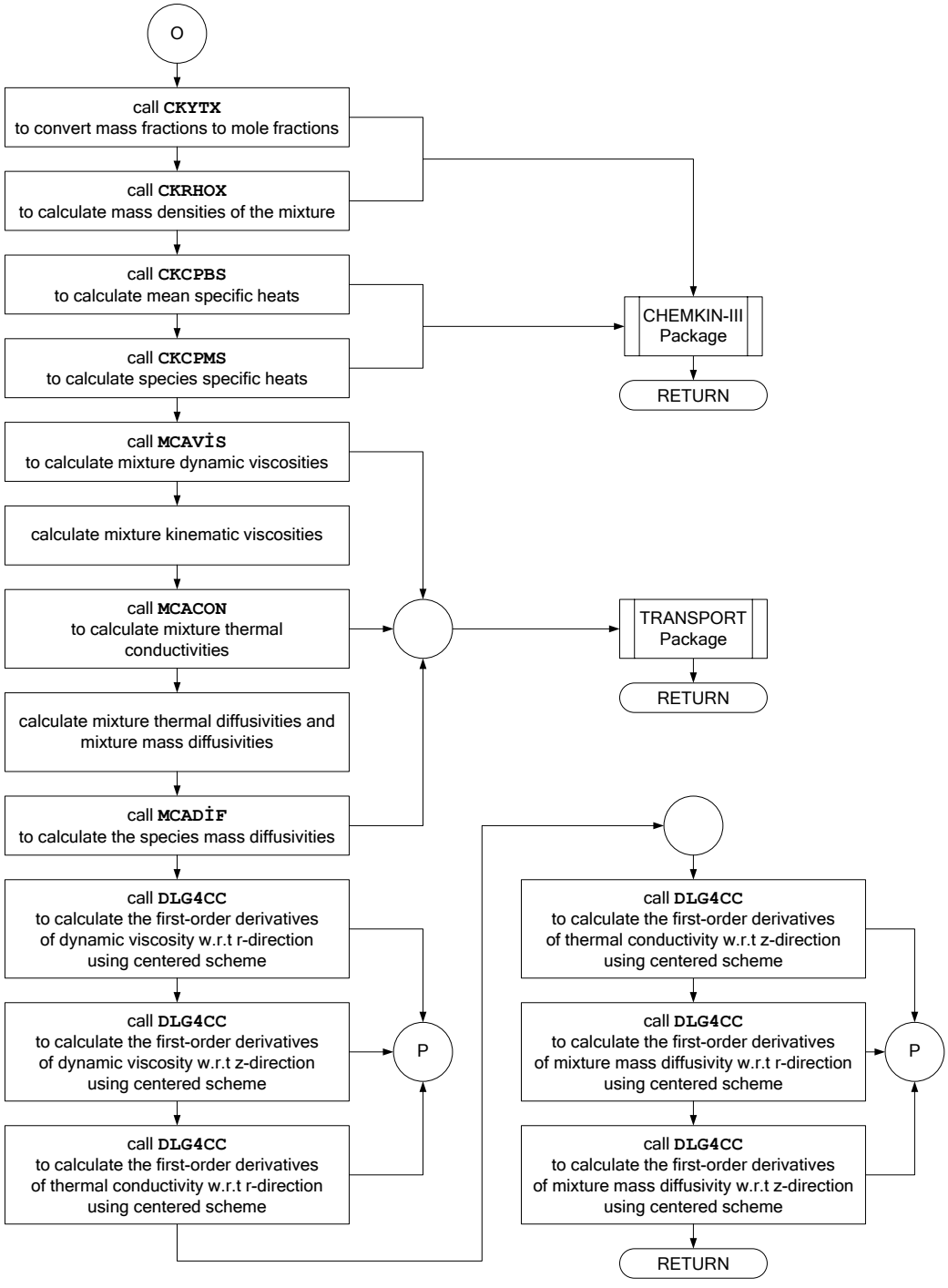


Figure 5.17: Algorithm of subroutine PROPERTY_ESTIMATION.

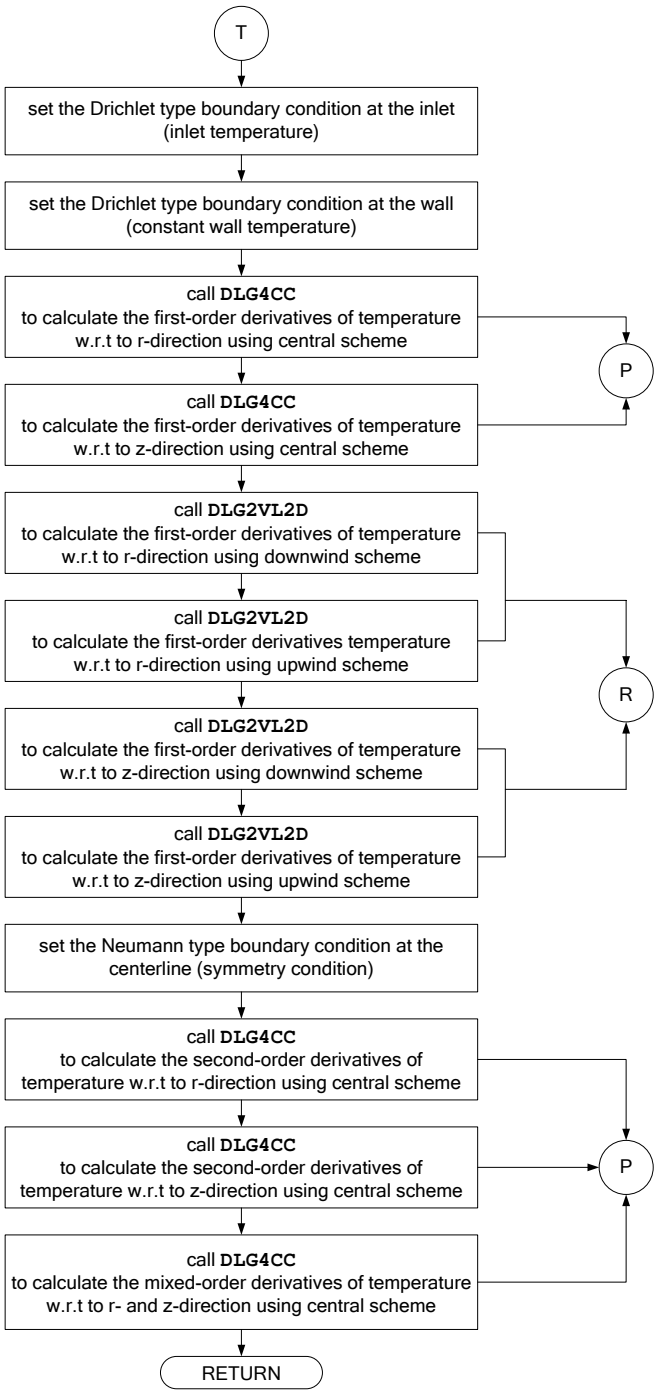


Figure 5.18: Algorithm of subroutine TEMPERATURE.

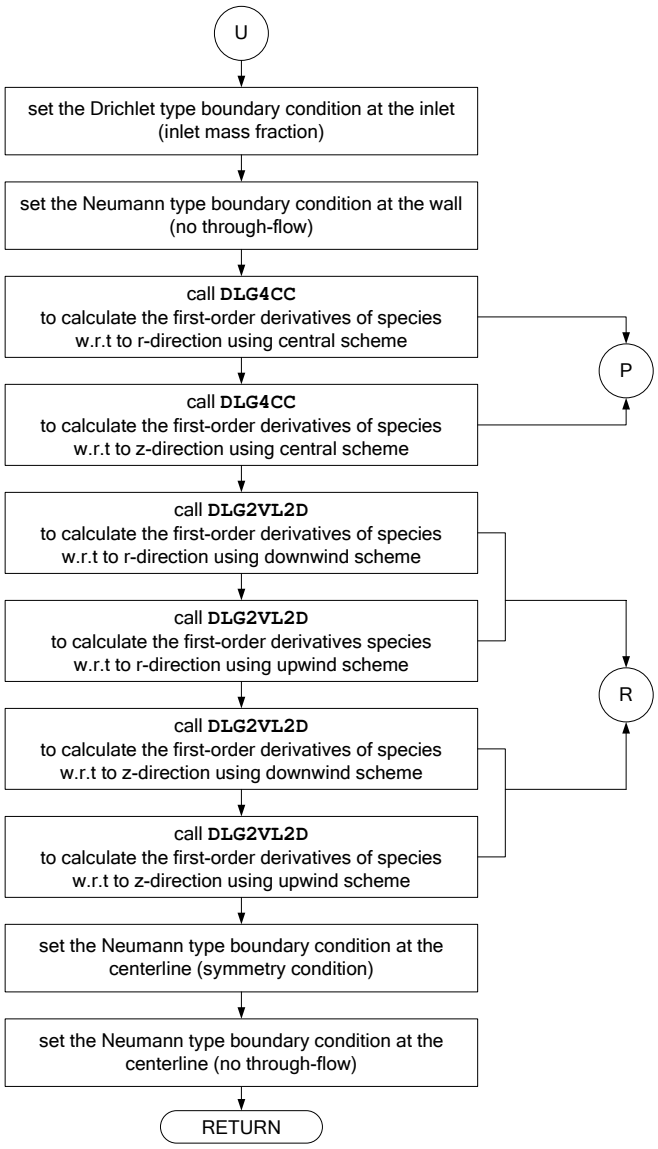


Figure 5.19: Algorithm of subroutine SPECIES.

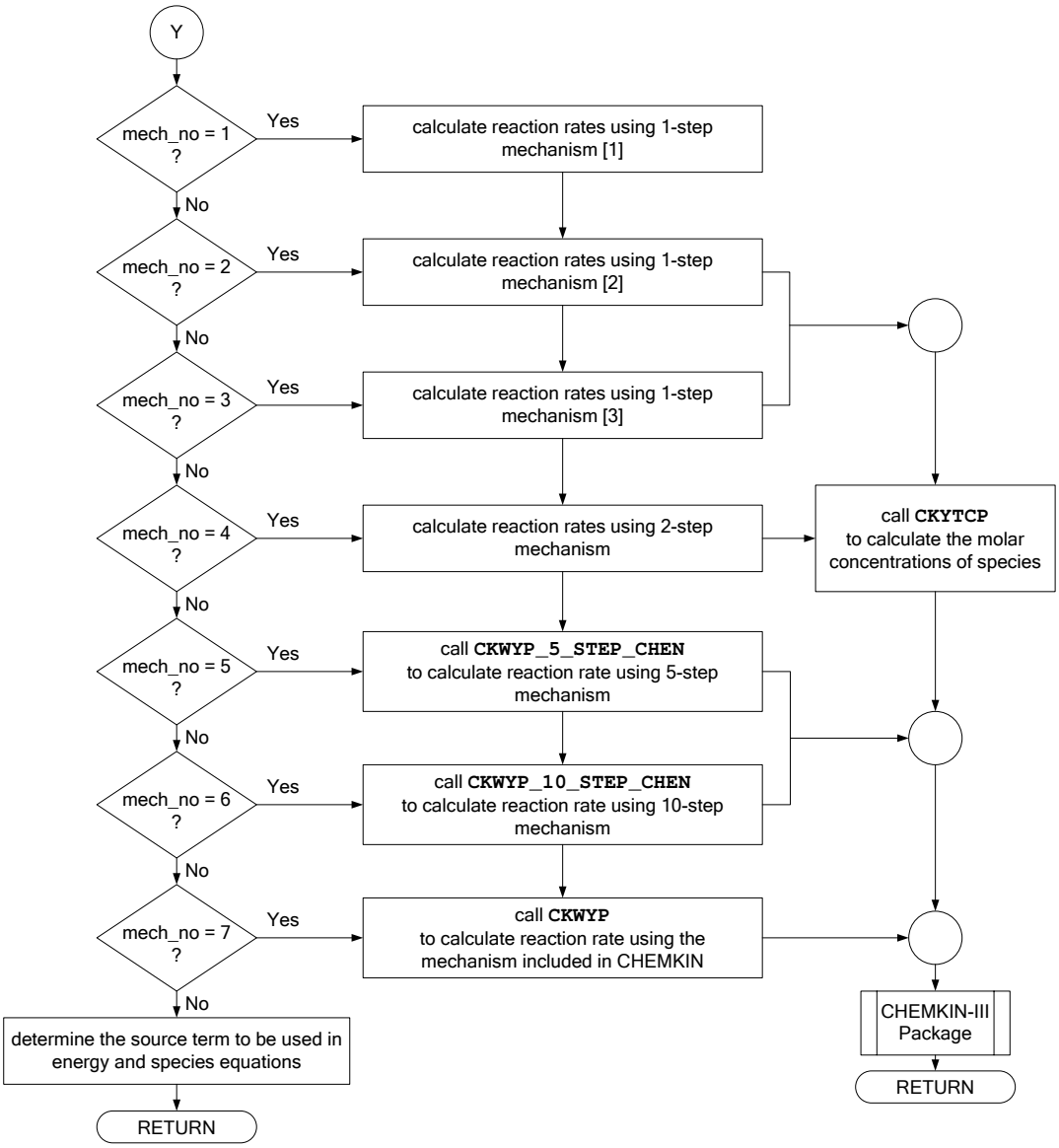


Figure 5.20: Algorithm of subroutine CHEMICAL_REACTION_MECHANISM.

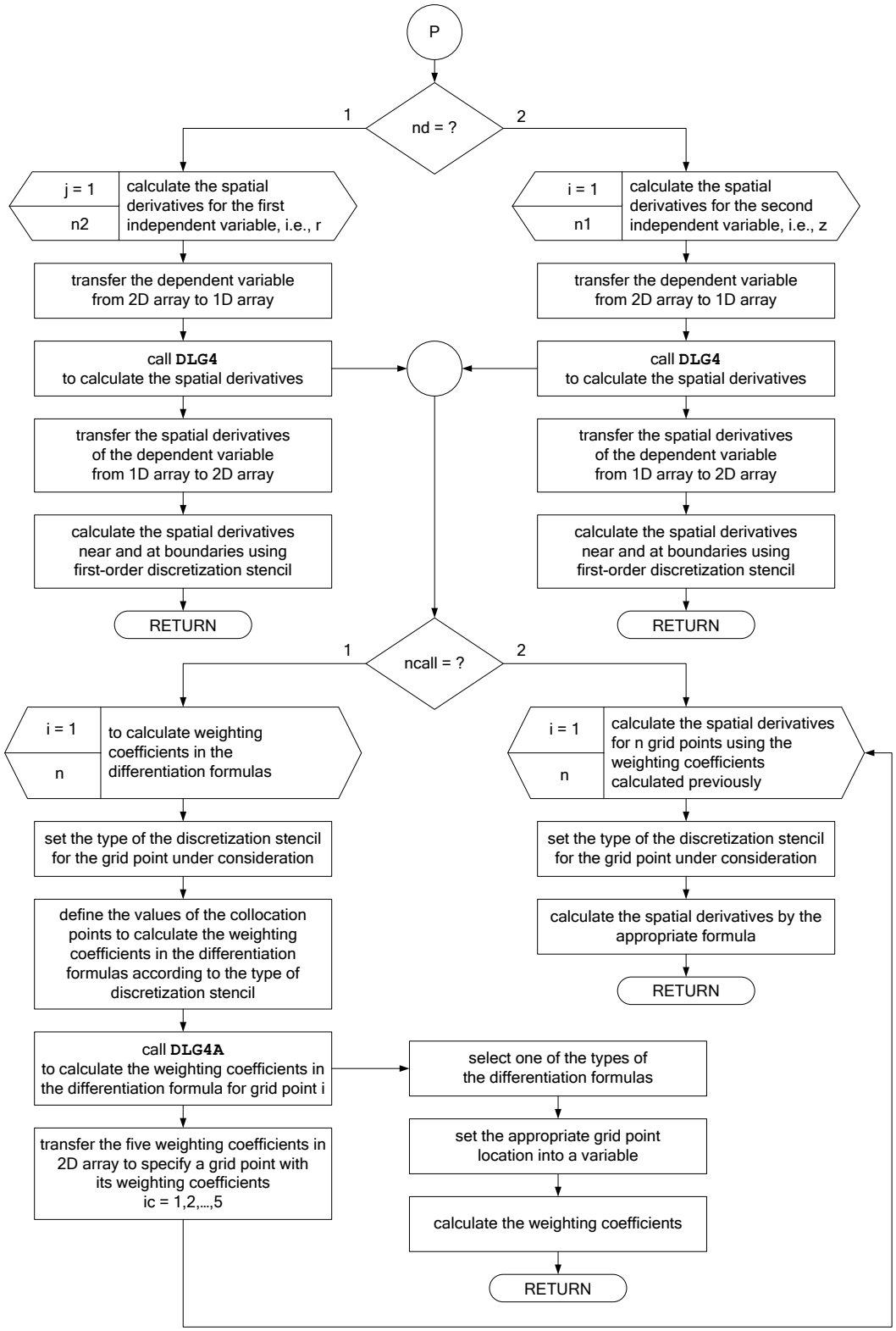


Figure 5.21: Algorithm of subroutine DLG4CC.

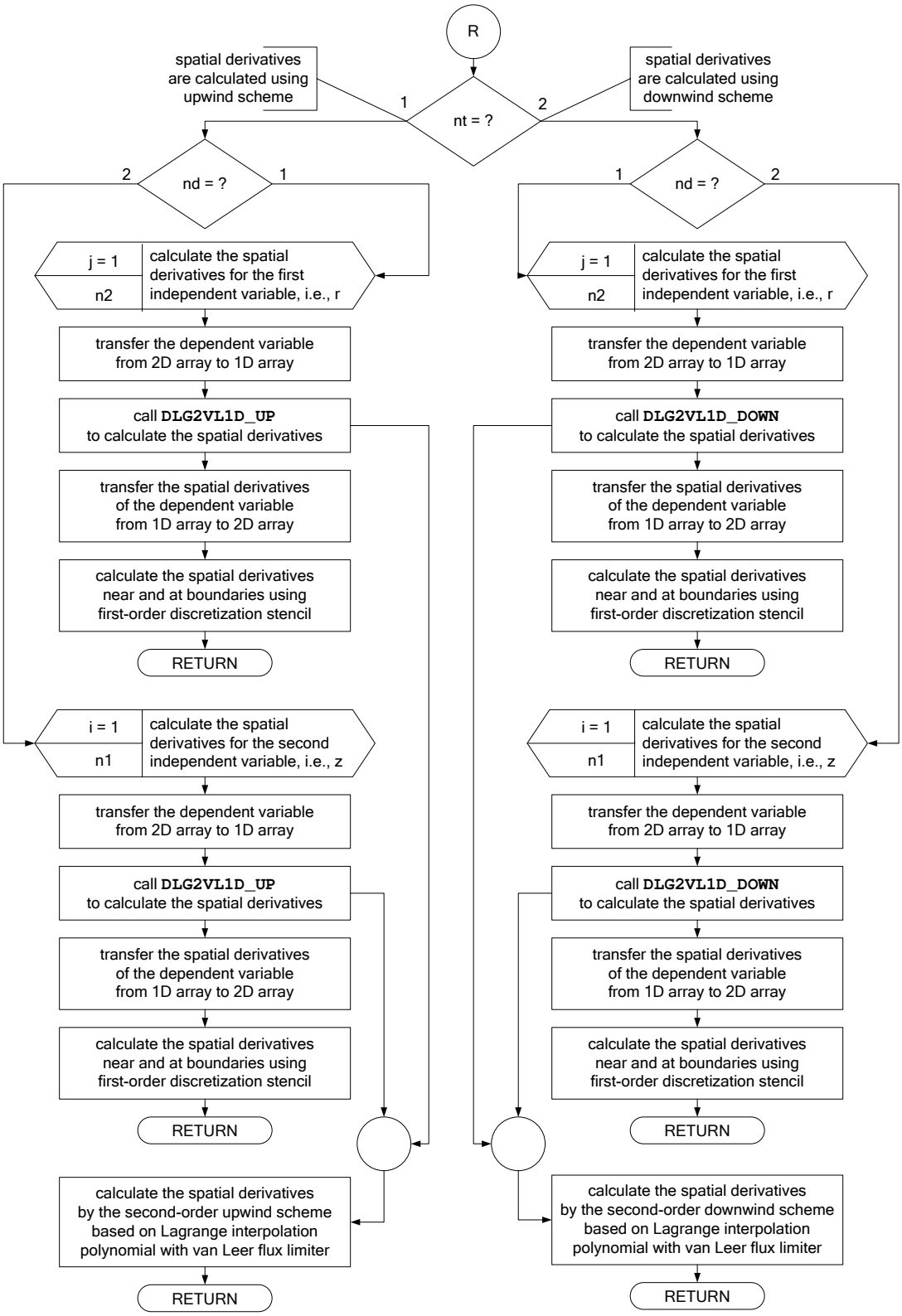


Figure 5.22: Algorithm of subroutine DLG2VL2D.

5.5 CHEMKIN and TRANSPORT Packages

Present algorithm uses CHEMKIN [55] and TRANSPORT [56] packages for evaluation of chemical reaction rates, thermodynamics and transport properties. The necessary information about the species and reactions should be provided prior to running the code to be able to use packages. The CHEMKIN Interpreter is a program which first reads the user-supplied symbolic description of the species and reactions in the problem from the input file *chem.inp*, a sample of which is given in Appendix B.1. Then it extracts the appropriate thermodynamic information for the species involved from the Thermodynamic Database file (*therm.dat*) which contains polynomial fits of thermodynamic information for many species, most of which are not needed for any given problem. The extracted information is stored in a file called Linking File (*chem.asc*) to be used by CHEMKIN library subroutines. The next program to be executed is the TRANSPORT Interpreter. The inputs required by the interpreter is supplied via *chem.asc* and TRANSPORT Database file (*therm.dat*) which contains molecular parameters for a number of species. Like the CHEMKIN Interpreter, it produces a Linking File (*tran.asc*) that is needed by the TRANSPORT package.

Both CHEMKIN and TRANSPORT subroutine libraries are initialized before being used in the code. This is accomplished by the subroutine CHEMKIN_INITIATION (Figure 5.6) . The Linking Files *chem.asc* and *tran.asc* are read in this subroutine to create data arrays for use internally by the subroutines in CHEMKIN and TRANSPORT libraries.

5.6 Pre- and Post-Processing

Information on the number of grid points to be used, dimensions of the computational domain, initial and boundary flow conditions, pressure equation and ODE solver related parameters, output locations radiation and chemistry model to be used and paths to the linking files of CHEMKIN and TRANSPORT packages are supplied to the code by means of input data files. Grid points at the boundaries and the size of the computational domain are described in *meshgen.ini* given in Appendix B.2. The

output locations at which the program is asked to produce radial and axial profiles of the dependent variables are specified in *locations.ini* (see Appendix B.3). All other inputs mentioned above including path to the *chem.asc* and *tran.asc* files are made through the file called *data.ini*, a sample of which can be found in Appendix B.4.

During the execution, the code produces two different types of raw output files; one in the form of snapshots at user defined multiples of the time-step and the other that contains the complete time history of the transient solution. In order not to restrict the code to a specific test case and preserve its generality, the processing of the raw output files is performed outside the code by means of separate FORTRAN programs COMBINE and EXTRACT given in Appendices C.1-C.2. The former is used to produce the transient output file which contains information up to any instant within the time range of the solution. The axi-symmetric mirror image of the two-dimensional fields and input data for future executions can also be obtained by this program. The latter is utilized to extract radial profiles of the dependent variables at the locations specified in *locations.ini*. The outputs obtained as result of processing are compatible with the industry's leading CFD visualization software TECPLOT 10 [81].

5.7 Programming Language and Compilation of the Code

The programming language used in the present study is FORTRAN 90 which is an efficient language for engineering purposes. The readily available library routines which are written in FORTRAN 77 were used as obtained without transforming to FORTRAN 90. Dynamic array allocation was used throughout the code in order to establish a proper foundation for its future parallelization. The program was designed to run on any LINUX or WINDOWS platforms without necessitating any modifications. The performance of the code was tested with major compiler distributions available such as Intel Fortran Compiler (versions 8.0, 9.0) and Compaq Visual Fortran (versions 6.5, 6.6).

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Preamble

The development of the computer code under consideration can be viewed as a two-stage process consisting of the following phases;

- Development of a non-iterative pressure based algorithm for the solution of momentum and energy equations,
- Incorporation of the solution of species and radiative transfer equations to the developed algorithm.

Upon completion of each stage, the predictive performance of the code was evaluated by applying it to a test case and validating its predictions against measurements and numerical solutions available in the literature.

The test case selected to serve as a benchmark for the first stage was the turbulent gas flow in a circular tube with strong wall heating studied by Shehata and McEligot [47]. The mean axial velocity and temperature predictions of the code was compared against the experimental and numerical data available in the literature. The ability of the code to predict transient non-isothermal internal flows was demonstrated on this test case.

The second test case, which serves as a benchmark for reacting radiating flows, was the laminar methane-air diffusion flame studied by Mitchell [45]. The predictions of axial and radial velocity, temperature and major species concentrations; *i)* in

the absence of a radiation model; *ii)* with gray radiation model; *iii)* with non-gray radiation model, were validated against the experimental measurements and numerical solutions available on the test case. The transient solutions produced by the code for reacting radiating flows were also presented.

All simulations were carried out on a personal computer with Pentium IV 3.0 Ghz processor having 2 Gb of RAM and running with Fedora Core 3.0 (Linux kernel 2.6.5) operating system.

In what follows, the description of the test cases and the numerical results obtained will be presented.

6.2 Test Case 1: Turbulent Gas Flow in a Circular Tube With Strong Wall Heating

Test case one involves the experimental investigation of turbulent flow of air in a vertical pipe with strong wall heating studied by Shehata and McEligot [47]. The test rig shown schematically in Figure 6.1 consists of a vertical, resistively heated circular test section of length L_t exhausting directly to the atmosphere and preceeded by a unheated entry region of length L_d for flow development. Air enters the development section with a flat velocity profile and a fully developed turbulent profile is obtained at the start of the test section. The experimental conditions were designed to approximate a uniform wall heat flux to air entering the test section at a uniform temperature.

A single hot wire sensor was utilized to measure the mean streamwise velocity and temperature at three axial locations (z/D = 3.2, 14.2 and 24.5) along the heated test section. The probe was employed as a hot wire for velocity measurements and as a resistance thermometer for pointwise temperatures. Convective and radiative heat losses were reduced by insulating the tube with a thick layer of silica bubbles, surrounded in turn by electrical heating tapes for guard heating. Details of the experiment and the measurements can be found elsewhere [82]. Geometrical parameters of the system and operating conditions used in the numerical simulations

are given in Figure 6.1. It should be noted that the length of the unheated entry region for fully developed profile utilized in this study was obtained from trial runs with various entry lengths.

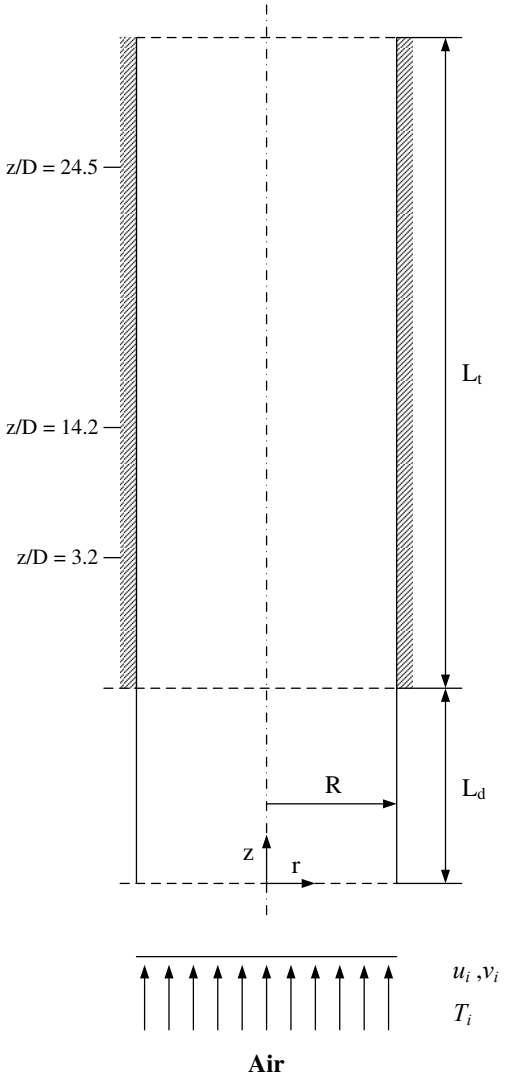
6.2.1 Grid and Time Step Sensitivity Study

In order to investigate the effect of grid density on the numerical results, the code was executed with three different set of grid points. In ascending order, these are ; 65×129 (set 1), 65×161 (set 2) and 65×513 (set 3) where in each pair the first and second numbers correspond to the number of grid points in r and z directions, respectively. Figure 6.2 shows the comparison of the mean axial velocity and temperature profiles at z/D = 14.2 obtained with the abovementioned resolutions. As can be seen from the figure, the results produced with all three resolutions are almost identical. Moreover, the CPU time requirement of the second and third sets are 1.37 and 5.94 times that of the first set, respectively. Therefore, 65×129 number of grid points was selected as the grid resolution to be employed in the computations.

The time step to be used for stable time-dependent computations was determined by running the code with three different time steps; $\Delta t = 1 \times 10^{-3}$ s, $\Delta t = 1 \times 10^{-4}$ s and $\Delta t = 1 \times 10^{-5}$ s and with a resolution of 65×129 grid points. Evaluation of the results shows that it was not possible to capture the unsteady nature of the flow by using $\Delta t = 1 \times 10^{-3}$ s. Furthermore, utilization of $\Delta t = 1 \times 10^{-5}$ s resulted in an oscillatory pressure field compromising the stability of the solution. It was $\Delta t = 1 \times 10^{-4}$ s which made it possible to obtain successful results by surmounting the problems associated with the former two time steps. Thus $\Delta t = 1 \times 10^{-4}$ s was designated as the time step to be used in the calculations.

6.2.2 Steady State Results

Comparison of the radial profiles of the mean axial velocity predicted by the present study, the experimental measurements [47], the DNS solution by Satake *et al.* [48] and the predictions of the previously developed MOL based CFD code [43] at three axial locations are demonstrated in Figure 6.3. As can be seen from the figure, the



***Geometrical Parameters*:**

R = 1.37 cm

L_d = 20.0 cm

L_t = 80.0 cm

***Operating conditions*:**

Reynolds number based on inlet: Re ≈ 4300

Pressure at the outlet: $p \approx 1$ atm

Wall temperature:

i) development section: T_w = 298 K

ii) heated test section: $T_w = T(z)$

***Inlet stream*:**

Temperature: T_i = 298 K

Axial velocity: u_i = 269.8 cm/s

Radial velocity: v_i = 0.0 cm/s

Figure 6.1: Schematic representation of the test rig and operating conditions.

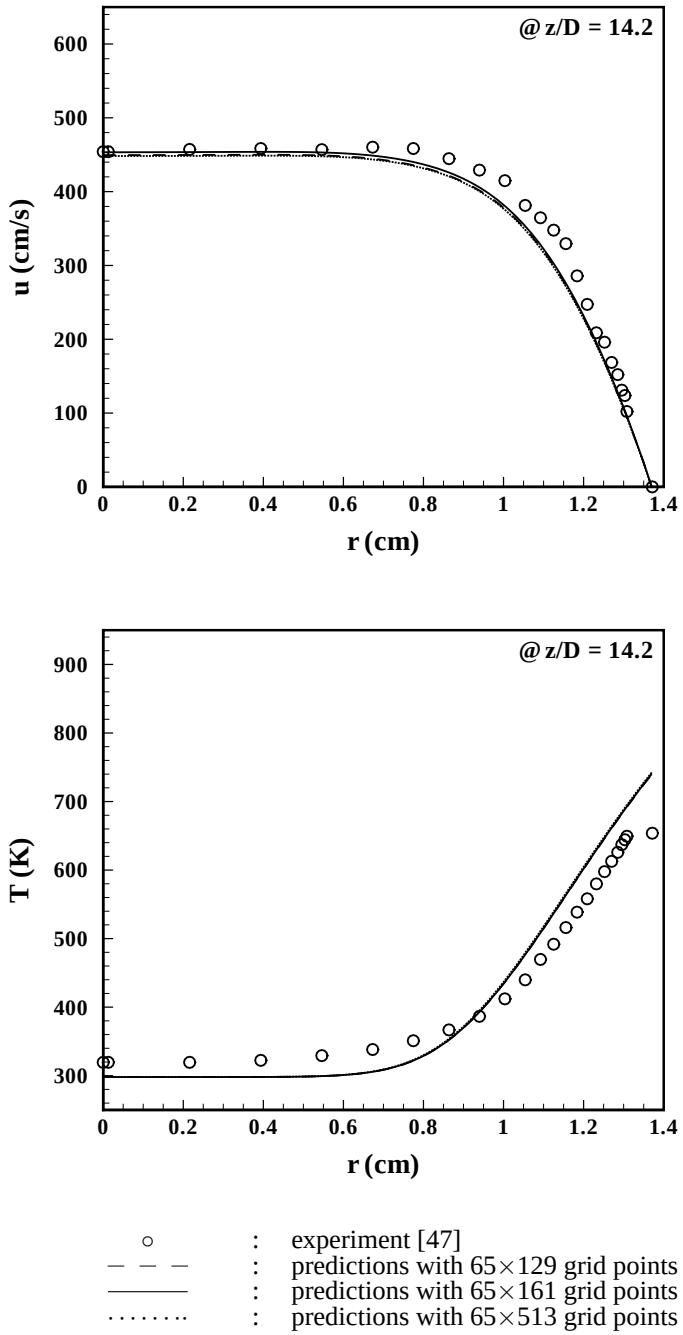


Figure 6.2: Radial profiles of mean axial velocity and temperature at z/D = 14.2 computed with different grid resolutions.

present predictions compare favorably with the measurements and the DNS data, particluarly at the first two axial locations, z/D = 3.2 and z/D = 14.2. The discrepancy between the predictions and measurements increases towards the centerline at the last station, z/D = 24.5, reaching a value of 7% of the maximum measured velocity. Nonetheless, the near-wall behavior is well captured at all locations, an achievement which was not possible with the previously developed code [43].

The mean temperature profiles are illustrated in Figure 6.4. Figure shows that the agreement between the predictions of this study and measurements is quite satisfactory at all locations along the pipe. Moreover, the experimental trends are mimicked almost exactly at the last station which could not have been achieved for the velocity profile at the same location. Although this may seem as a contradiction at a first galnce, considering the fact that heat transfer in the vicinity of the wall is mostly governed by conduction and velocities at this region is much lower compared to the centerline due to the thickening of the boundary layer, it is not surprising to observe such trends.

The axial pressure distribution at the centerline is presented in Figure 6.5. Figure displays that the pressure exhibits a linear trend and becomes equal to atmospheric pressure at the end of the test section. The maximum discrepancy between the predictions and the measurements occurs towards the inlet. This is typical of pressure based algorithms employing Neumann type boundary condition at the inflow for the solution of pressure equation.

The general structure of the flow under consideration is illustrated by means of steady-state contours of axial and radial velocity, temperature and pressure in Figure 6.6.

It can be seen from the temperature contours that as a result of adiabatic heating, an increasing temperature profile develops at the wall reaching to a maximum of 820 K at the exit. Consequently, the amount of heat penetrating towards the centerline increases downstream. This effect tends to decrease the densities and in turn leads to flow acceleration as depicted by the axial velocity contours. The Reynolds number

at the exit is approximately 3300 which points out to a laminarized flow [47, 83, 84] considering that the inlet Reynolds number is 4300. However, flat velocity profile characteristic to turbulent flows is still visible at the outlet. The pressure contours display that no significant change occurs in radial direction as one would expect for a typical pipe flow. Therefore, variation of radial component of velocity is limited to a narrow range (-8.0 to 0.2 cm/s) as illustrated by radial velocity contours.

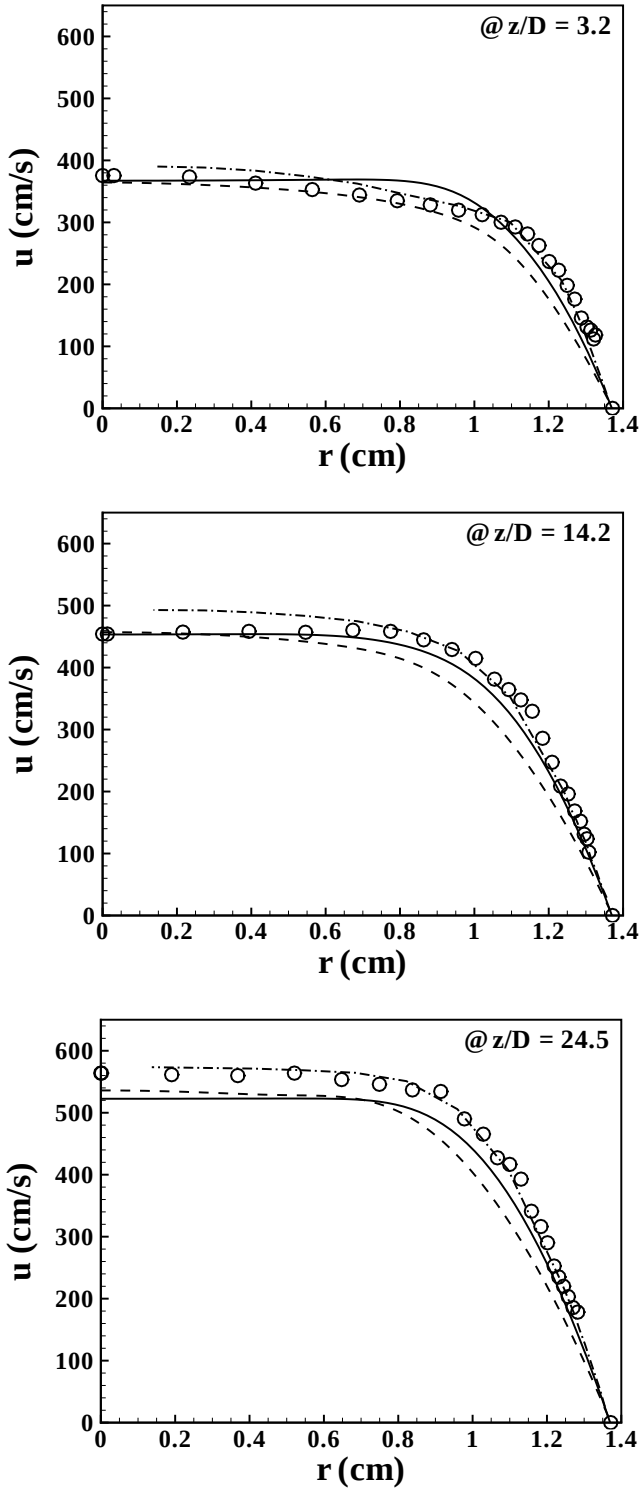


Figure 6.3: Radial profiles of mean axial velocity at three axial locations.

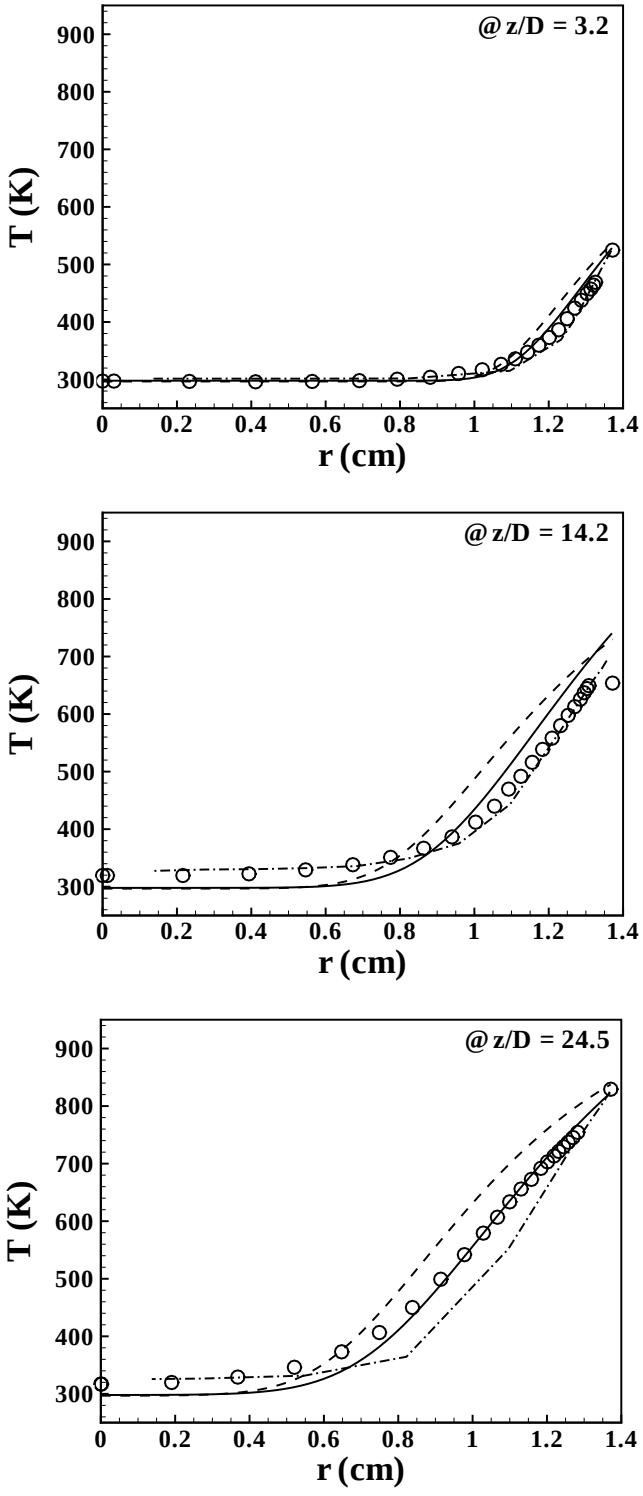


Figure 6.4: Radial profiles of temperature at three axial locations.

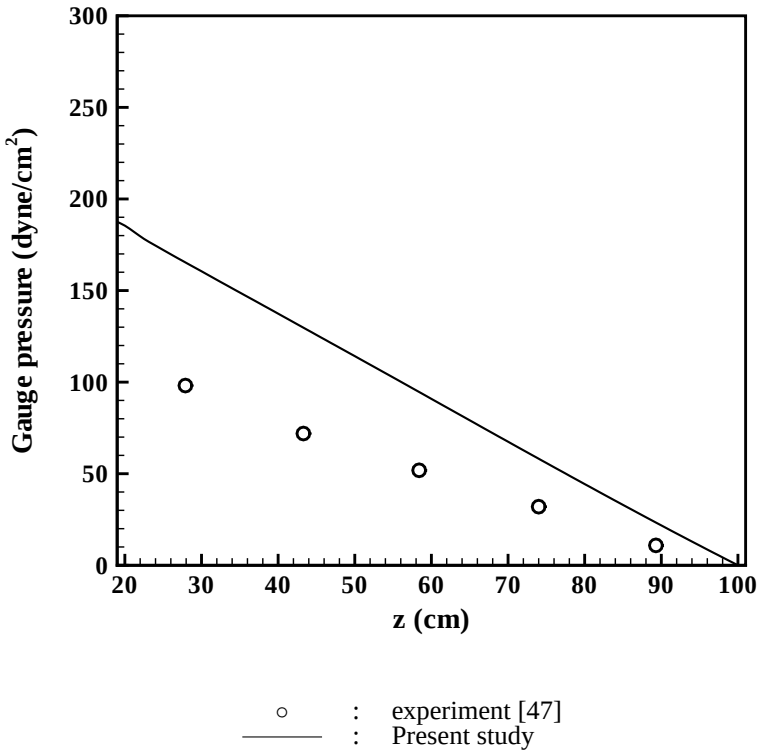


Figure 6.5: Axial pressure distribution along the centerline.

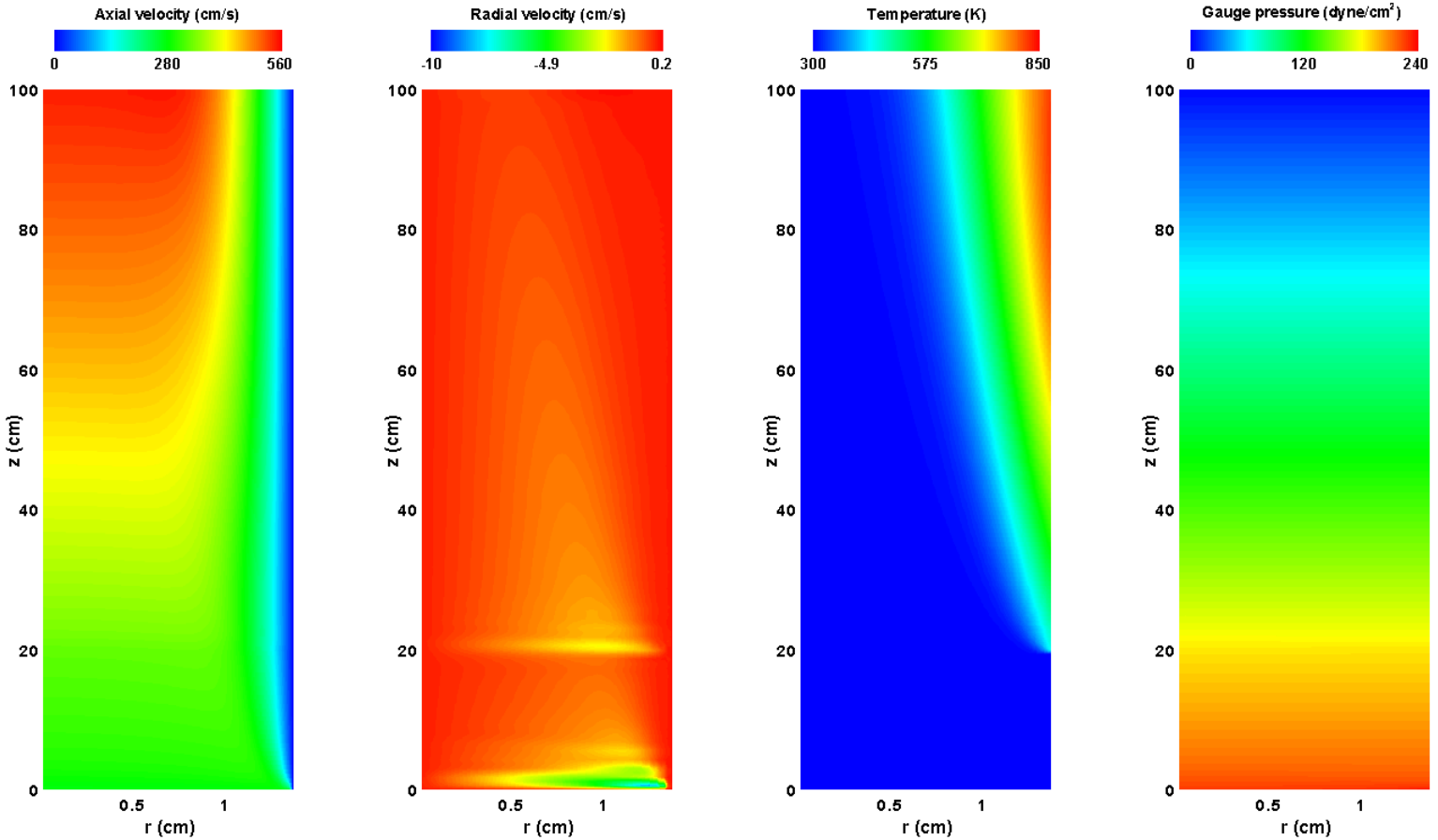


Figure 6.6: Contours of axial and radial velocity, temperature and pressure.

6.2.3 Transient Results

In order to demonstrate the predictive ability of the present code for transient solutions, it was executed for a final time of $t_f = 0.4$ s with a time step of $\Delta t = 1 \times 10^{-4}$ s. The executions took about 1920 s for 65×129 grid points.

Figures 6.7 shows the time development of the mean axial velocity field by color contours. The flow is initially at rest. As soon as the flow initiates, a velocity boundary layer starts to form near the solid surface. As time progresses, the boundary layer gets thicker and velocity increases in the core region due to the increase in temperature. Despite the laminarizing effect of increasing temperature, flat velocity profile typical to turbulent flows remains persistent at the outflow.

Time development of the mean temperature field is illustrated in Figure 6.8. It can be seen from the figure that, the thermal boundary layer thickness increases with time as the amount of energy penetrating towards the centerline increases and at steady-state it covers almost three fourths of the radius. Unlike the velocity field exhibiting turbulent characteristics, the temperature field at the outflow resembles laminar flow with a nearly parabolic profile. This can be attributed to the extensive heating rate and relatively low Reynolds number combination used in the experiments which is classified as being the borderline between laminar and sub-turbulent regime [47].

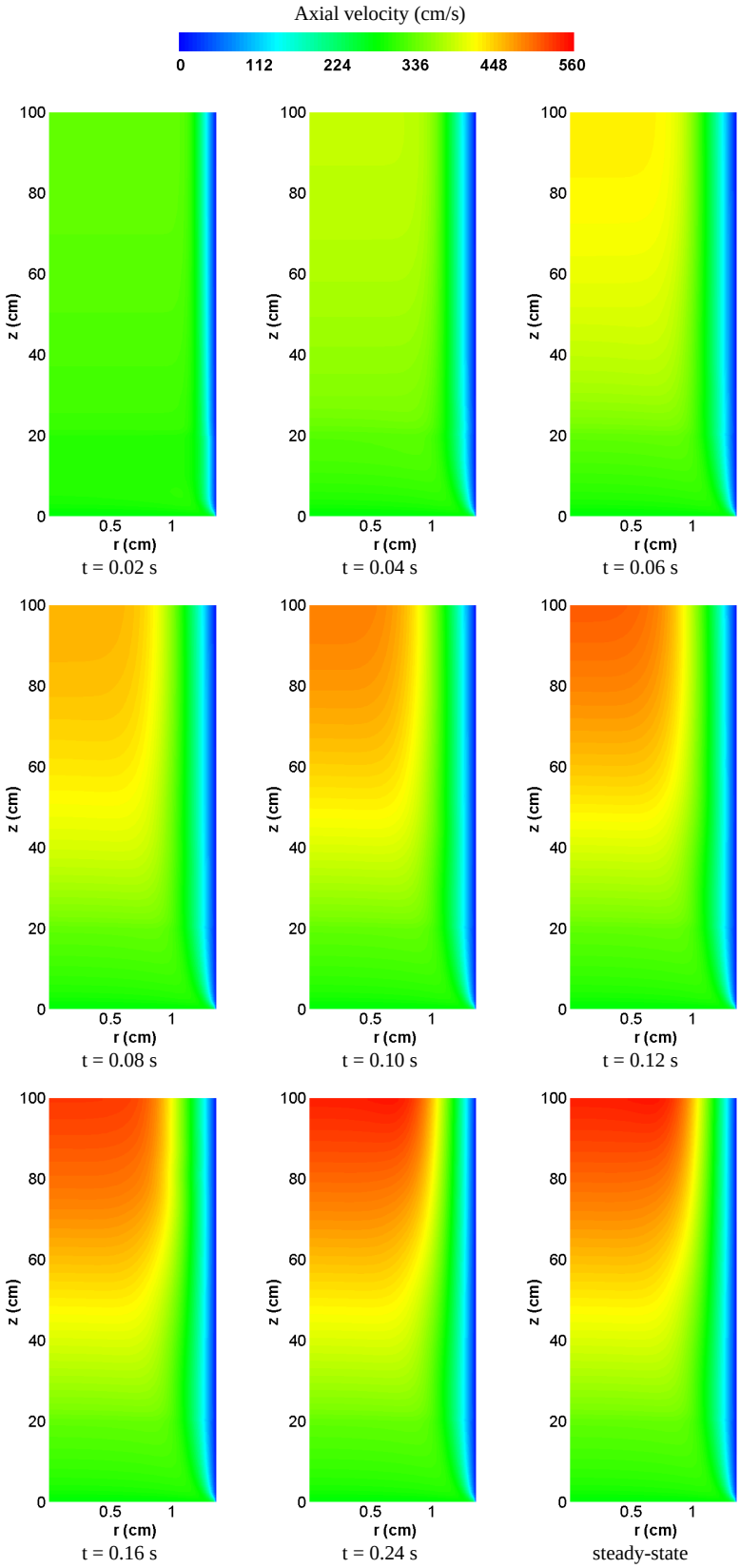


Figure 6.7: Time development of axial velocity field.

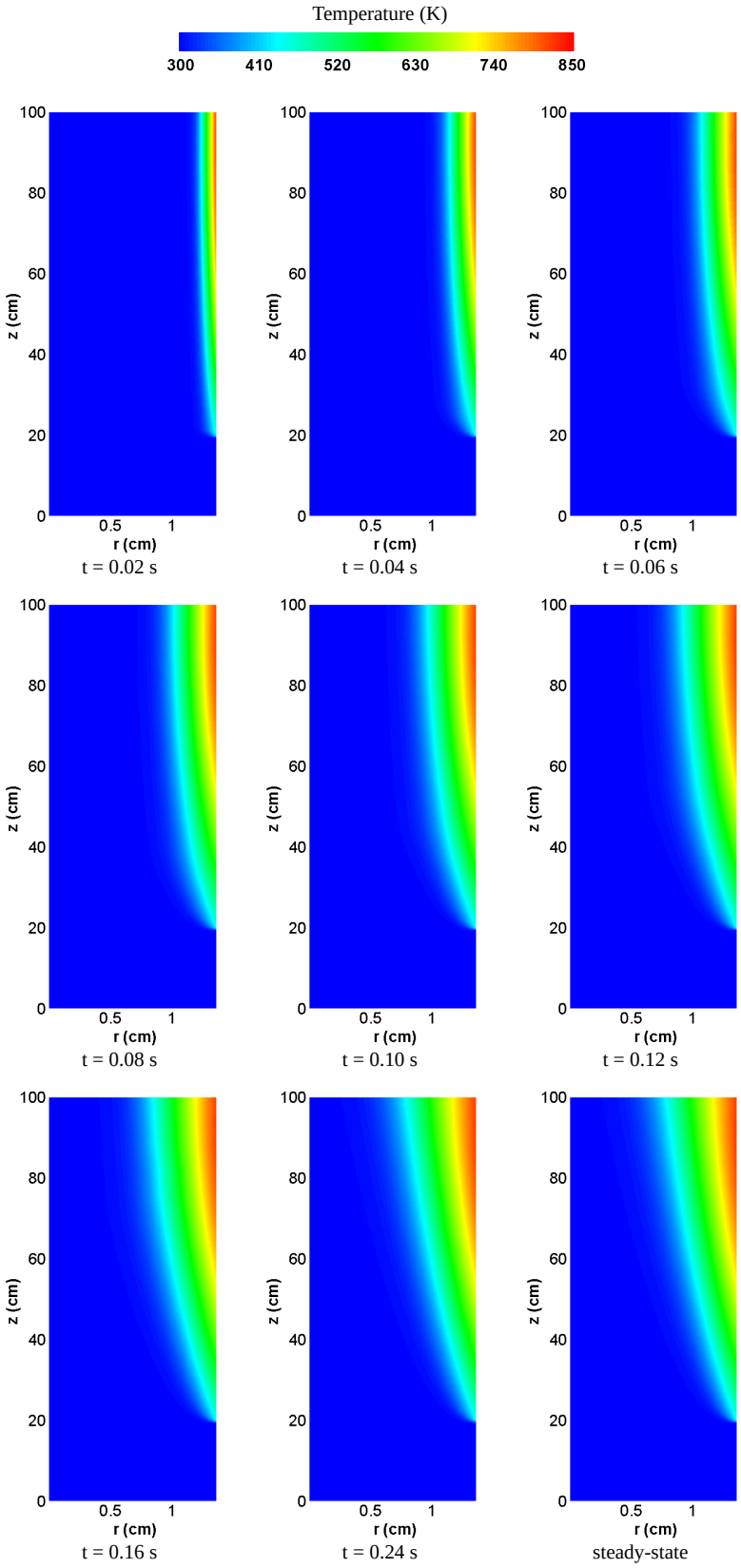


Figure 6.8: Time development of temperature field.

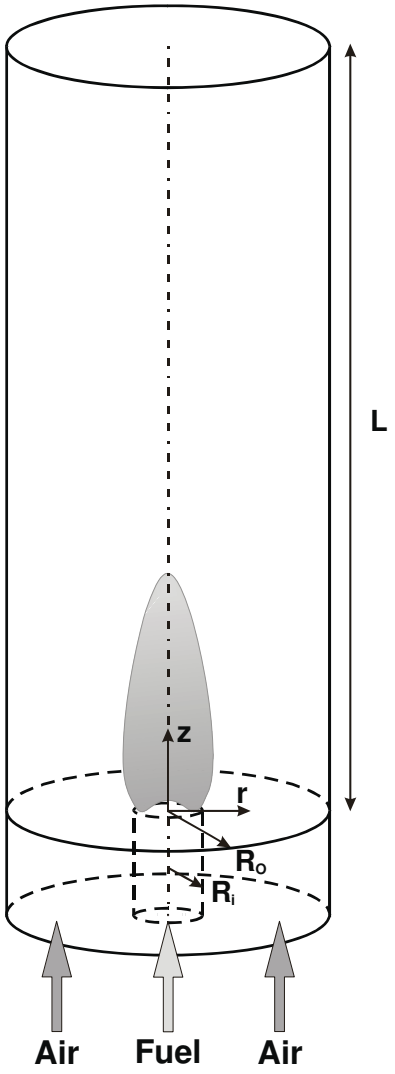
6.3 Test Case 2: Laminar Methane-Air Diffusion Flame

The problem to be investigated is a confined, steady, atmospheric, axi-symmetric, co-flowing laminar methane-air diffusion flame studied by Mitchell [45] in a vertical, cylindrical laboratory flame burner. The burner, schematically shown in Figure 6.9, consists of two concentric brass tubes of radii R_i and R_o. Fuel flows through the inner tube and air enters the system through the outer tube. Flat velocity profiles inside the tubes and uniform velocity distribution at the outlets are maintained by several screens packed inside the tubes made of perforated brass plate with high thermal conductivity is placed at the burner inlet to assure parallel inlet flows with uniform temperature profile. A pyrex glass cylinder is used to produce a confined flame and to define the boundaries of the combustion system.

Technical grade methane (98% purity) and compressed air are fed to the burner through their respective inlets. Critical orifices control the flowrates of the methane at 5.7 cm^3/s and air at 187.7 cm^3/s. Upon ignition, a cylindrical methane-air diffusion flame of definite shape and height is produced in the system. The resulting temperature and concentration profiles of CH_4, O_2, H_2O, CO, CO_2, NO, H_2 and N_2 at three axial locations above the burner ($z = 1.2$, 2.4 and 5.0 cm) and along the centerline ($r = 0$ cm) are measured. Temperature measurements were performed using platinum vs. platinum-13% rhodium thermocouples. Radiation and conduction corrections were made to determine local gas temperatures. Samples were withdrawn from the burner with a quartz microprobe and then analyzed using a gas chromatograph. Velocities were determined by a laser Doppler velocimetry seeding the flame with TiO_2 particles. Details of the experimental apparatus and analytical procedure are described in [45, 46]. Geometrical parameters and operating conditions used in the numerical simulations are shown in Figure 6.9

6.3.1 Grid and Time Step Sensitivity Study

The effect of grid density on the numerical results was evaluated by executing the code for three different set of grid points in the absence of radiative transfer model. In ascending order, these are ; 33×129 (set 1), 65×129 (set 2) and 65×161 (set 3),



Geometrical Parameters:

R_i = 0.635 cm

R_o = 2.54 cm

L = 30.0 cm

Operating conditions:

Pressure at the exit of the burner: p = 1 atm

Wall temperature: T_w = 298 K

Inlet streams:

i) Inner tube (Fuel side):

Inflow axial velocity: u_F = 4.5 cm/s

Inflow radial velocity: v_F = 0.0 cm/s

Temperature: T_F = 298 K

Composition: Y_{CH_4} = 1.0

ii) Outer tube (Oxidizer side):

Inflow axial velocity: u_A = 9.88 cm/s

Inflow radial velocity: v_A = 0.0 cm/s

Temperature: T_A = 298 K

Composition: Y_{O_2} = 0.232, Y_{N_2} = 0.768

Figure 6.9: Schematic representation of the co-flowing jet diffusion flame and operating conditions.

where in each pair the first and second numbers correspond to the number of grid points in r and z directions, respectively. Figure 6.10 shows the comparison of the axial velocity, temperature and mole fractions of CO_2 and H_2O profiles computed at $z = 2.4$ cm with the abovementioned grid resolutions. As depicted by the figure, increasing the resolution beyond 65×129 grid points has no effect on the results and profiles obtained with 65×129 and 65×161 grid points overlap each other. Hence, 65×129 number of grid points was selected as the grid resolution to be employed in the computations.

In order to determine the time step to be used for stable time-dependent calculations, the code was executed with three different time steps; $\Delta t = 1\times10^{-4}$ s, $\Delta t = 1\times10^{-5}$ s and $\Delta t = 1\times10^{-6}$ s and with a resolution of 65×129 grid points. Upon testing, it was seen that $\Delta t = 1\times10^{-4}$ s was not sufficient to overcome the stiffness brought by combustion and $\Delta t = 1\times10^{-6}$ s over-amplifies the source term of the pressure Poisson equation (Equation (4.23)), hence leading to unstable solutions. Using $\Delta t = 1\times10^{-5}$ s on the other hand, it was possible to capture the transient development and the physics of the flow without compromising the stability of the solution. Therefore, $\Delta t = 1\times10^{-5}$ s was designated as the time step to be utilized in the calculations.

6.3.2 Numerical Results Without Radiation

The code was executed in the absence of a radiation model for a suddenly started flow of fuel and air ignited at their intersection region for a final time of $t_f = 0.5$ s with the grid resolution and time step given above. Under these conditions, the executions took about 20 hours of CPU time and the steady-state results obtained are presented below.

The radial profiles of temperature, axial velocity, mole fractions of CH_4, O_2, CO_2, H_2O and N_2 at three axial locations above the burner inlet and axial profiles along the centerline predicted by the code are plotted together with the experimental data [45, 46] and the numerical solutions of Tarhan [1, 85] obtained with the previously developed code and a commercial CFD code (FLUENT) in Figures 6.11-

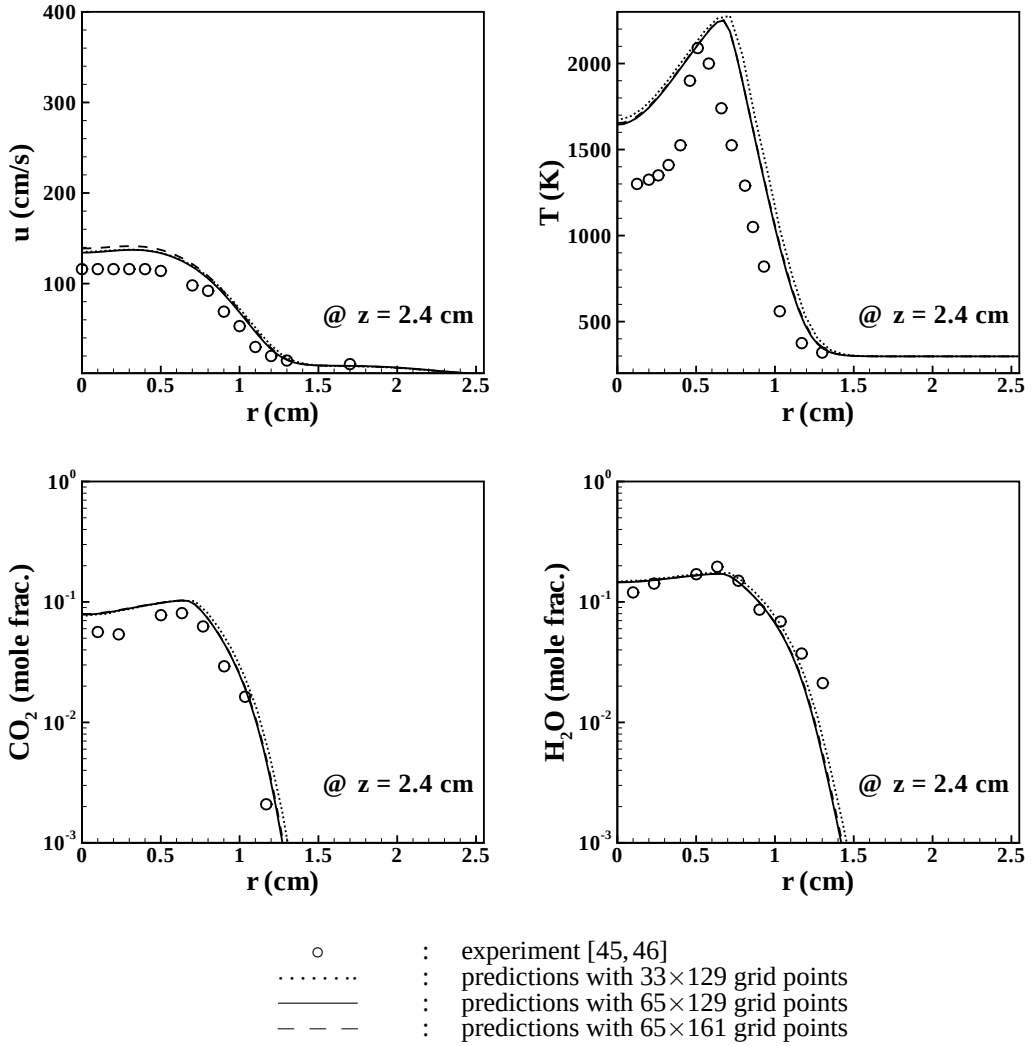


Figure 6.10: Radial profiles of axial velocity, temperature, mole fractions of CO_2 and H_2O at z = 2.4 cm computed with different grid resolutions.

6.17.

Figure 6.11 shows the radial and axial profiles of temperature. As can be seen from the figure, temperature inside the flame increases to a maximum at the flame front and decreases sharply in radial direction outside the reaction zone and towards the wall. The flame height, defined as the location along the centerline where the maximum temperature occurs, was computed as 7.4 cm, higher than the experimental value of 5.8 cm as displayed in the axial profile. Inspection of radial profiles reveals that the predictions at the first axial location follow the experimental data closely. However, temperatures are over-predicted, particularly inside the flame, as downstream locations are reached. This trend was also exhibited by the other two numerical solutions displayed in the figure. This and the discrepancy between the predicted and measured flame heights can be attributed to global reaction mechanism employed in all of the presented numerical solutions. Supporting evidence was found in a study by Tarhan [1,85] where it was shown that utilization of multi-step reaction mechanisms involving minor species such as CO and H_2 actually lowers the adiabatic flame temperature by lowering the total heat of reaction. In addition, it was also reported in the same study that the flame height predictions obtained with five- and ten-step reaction mechanisms were in excellent agreement with the experimental measurement.

The radial and axial profiles of axial velocity are illustrated in Figure 6.12. As can be seen from the radial profiles, the agreement between the predicted and measured velocities is closely related to the temperature predictions of the code. The experimental data is mimicked almost exactly at the first axial location and the agreement tends deteriorate at downstream locations. Apparently, this is due to the fact that over-predicted temperatures (owing to the reasons explained above) lead to under-predicted densities and in turn over-predicted velocities. Nonetheless, comparisons with measurements reveal that the predictive accuracy of the code is better than that of the previously developed code and comparable to that of the commercial one (FLUENT). The axial profiles plotted in the absence of experimental data clearly shows the acceleration of the flow due to high temperatures resulting

from the reaction.

Figures 6.13 show the radial and axial profiles of CH_4 mole fractions. As can be seen from the figure, predictions compare favorably with the measurements. The absence of measured data for methane at the axial location z = 5.0 cm indicates that methane is completely consumed before reaching this location. This trend was not accurately predicted in any of the presented numerical solutions. The axial profile along the centerline display that all methane is consumed within the range between the burner inlet and the flame height as expected.

Outside the flame, oxygen concentration is almost equal to its inlet value at all axial locations as depicted by Figure 6.14. The experimental measurements in radial direction show that the concentration decreases to a minimum value inside the reaction zone and then some of it is convected towards the centerline where temperatures are relatively lower (see Figure 6.11). The axial profile along the centerline also indicates the presence of oxygen inside the flame at lower heights. This behavior can not be observed in the present predictions as the oxygen is almost completely consumed inside the reaction zone owing to the over-predicted temperatures at this region of the burner. The absence of both the fuel and the oxidant inside the flame can be attributed to the global reaction mechanism employed [85]. On the whole, it can be said that the predictions agree reasonably well with the experimental data.

The profiles of major combustion products CO_2 and H_2O which are also significant in radiative heat transfer calculations due to their absorbing emitting nature are displayed in Figures 6.15 and 6.16. As depicted by the radial profiles, both species concentrations reach their peak values at the flame front where temperature is at its maximum and then start to decrease towards the wall. Along the centerline, both CO_2 and H_2O concentrations increase sharply inside the flame convected downstream. On the whole, the predictions and the experimental data are in excellent agreement.

Figure 6.17 illustrates the radial and axial profiles of N_2 mole fraction. As can be seen from the figure, nitrogen diffuses into the flame zone at all axial locations.

Axial profile shows that N_2 mole fraction increases steeply just above the burner inlet and attains its maximum after short distance from the inlet. Predictions are in good agreement with the experimental measurements.

Finally, the profiles of radial velocity at two axial locations above the burner inlet are presented in Figure 6.18. The predictions of the code are compared against other numerical solutions due to the absence of experimental measurements. As depicted by the figure, present predictions compare favorably with the predictions of Zhang [86] and FLUENT CFD code and they all follow the same trend in contrast to predictions of Tarhan [1] which deviate largely from these three in radial direction. This discrepancy will also be illustrated in contour plot of radial velocity.

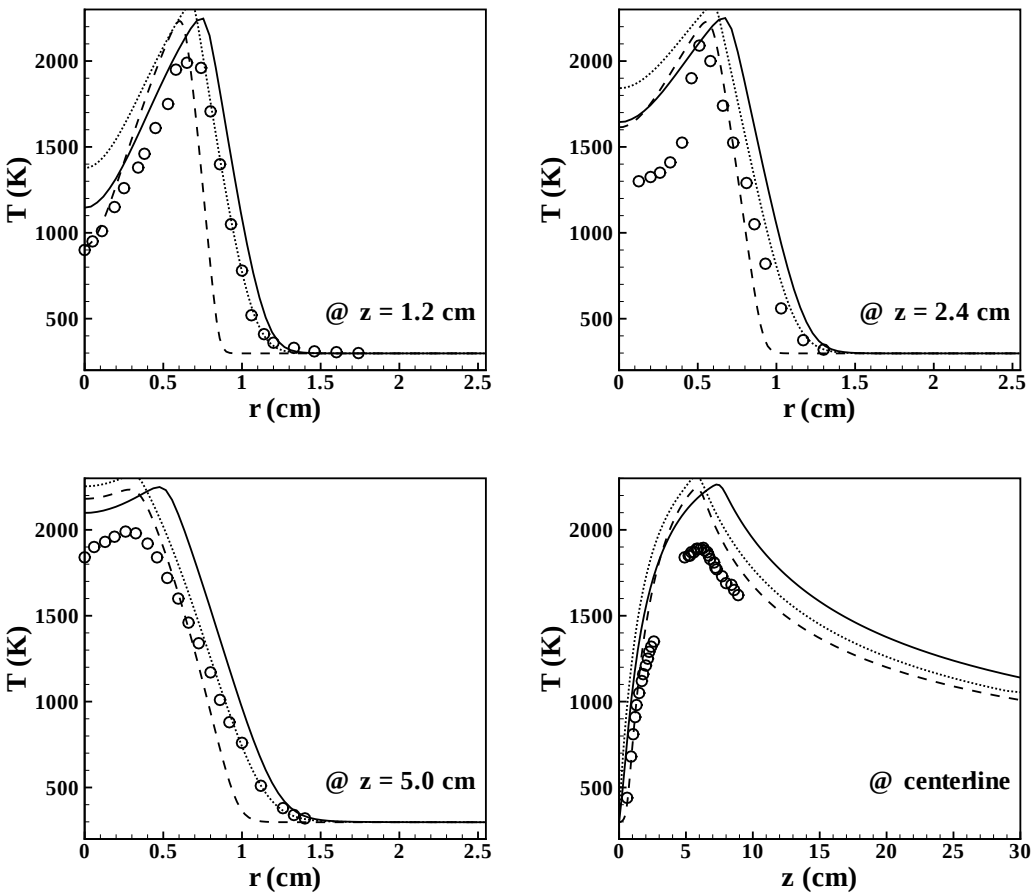


Figure 6.11: Radial and axial profiles of temperature at three axial locations and along the centerline.

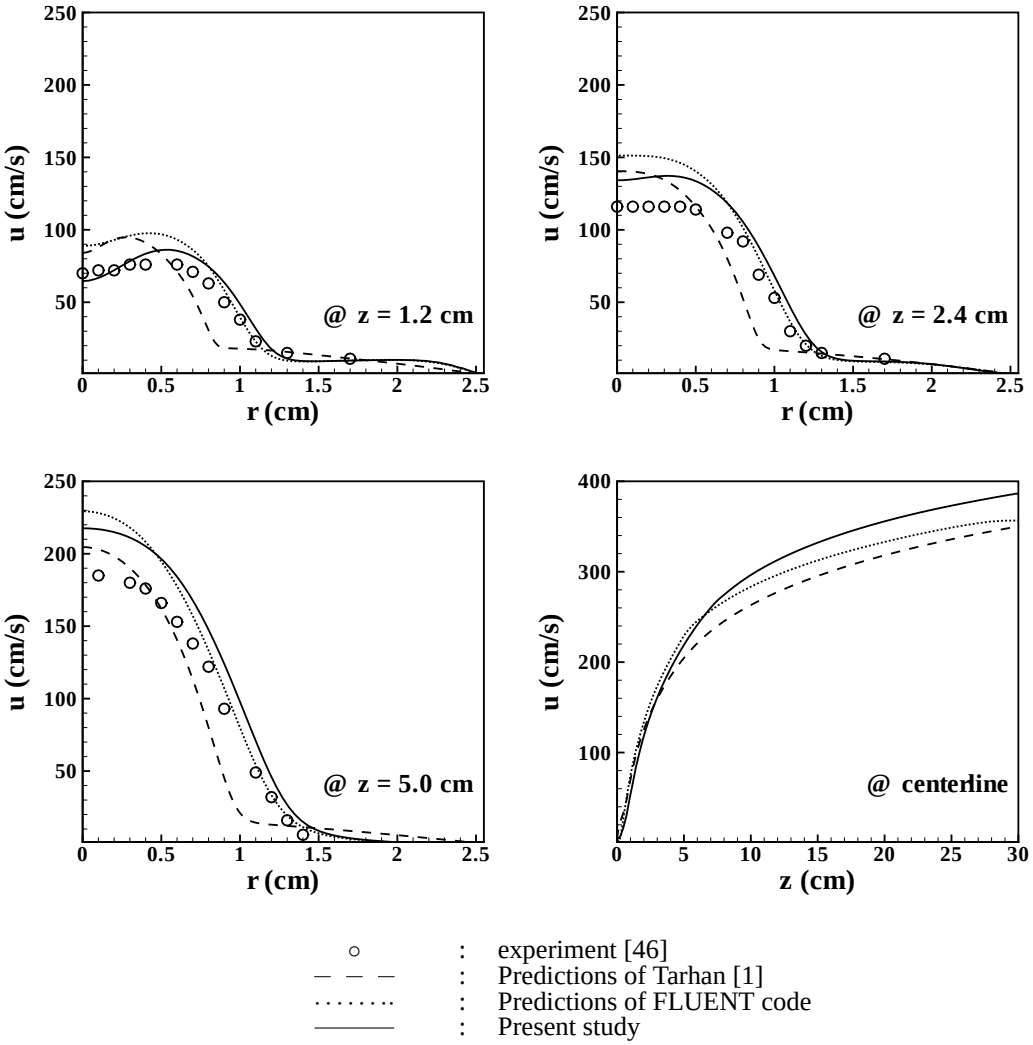


Figure 6.12: Radial and axial profiles of axial velocity at three axial locations and along the centerline.

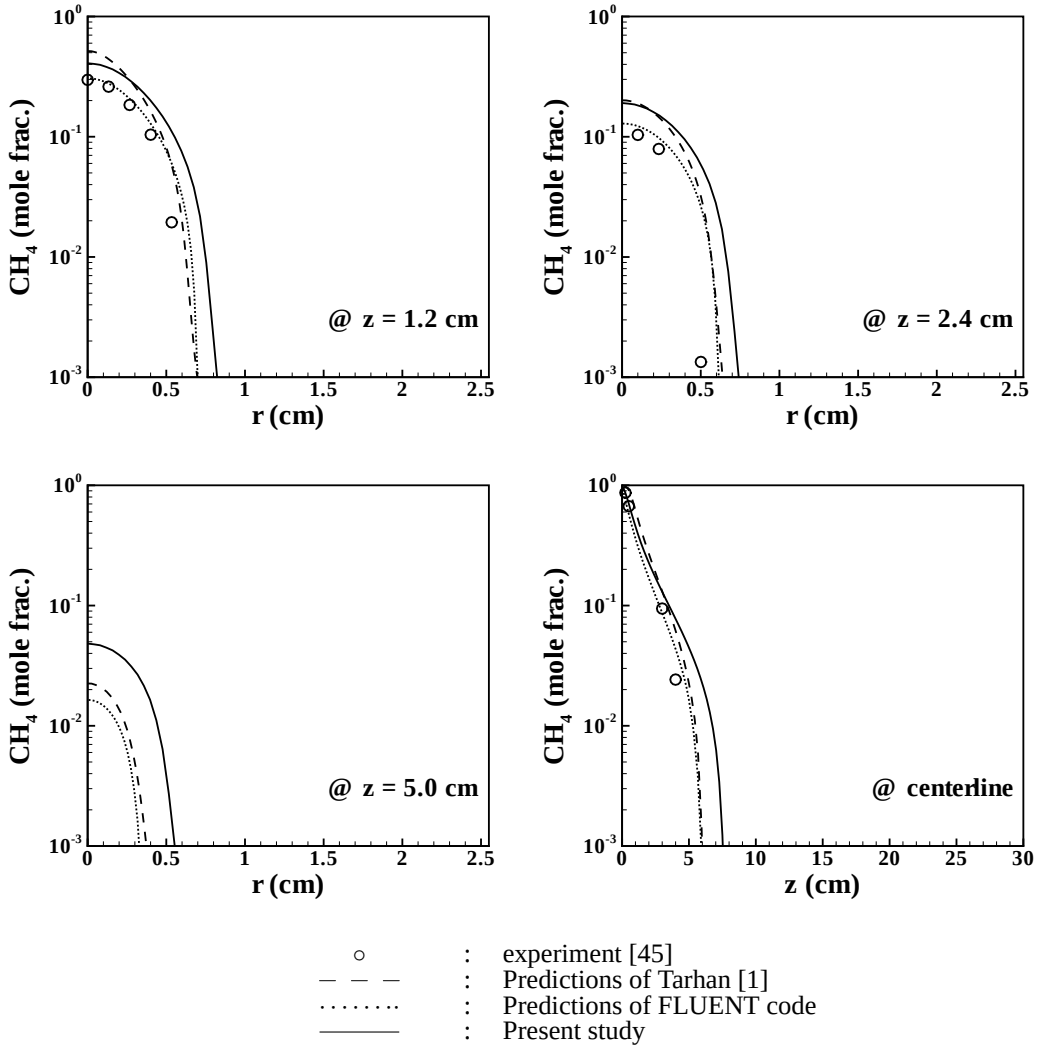


Figure 6.13: Radial and axial profiles of CH_4 mole fractions at three axial locations and along the centerline.

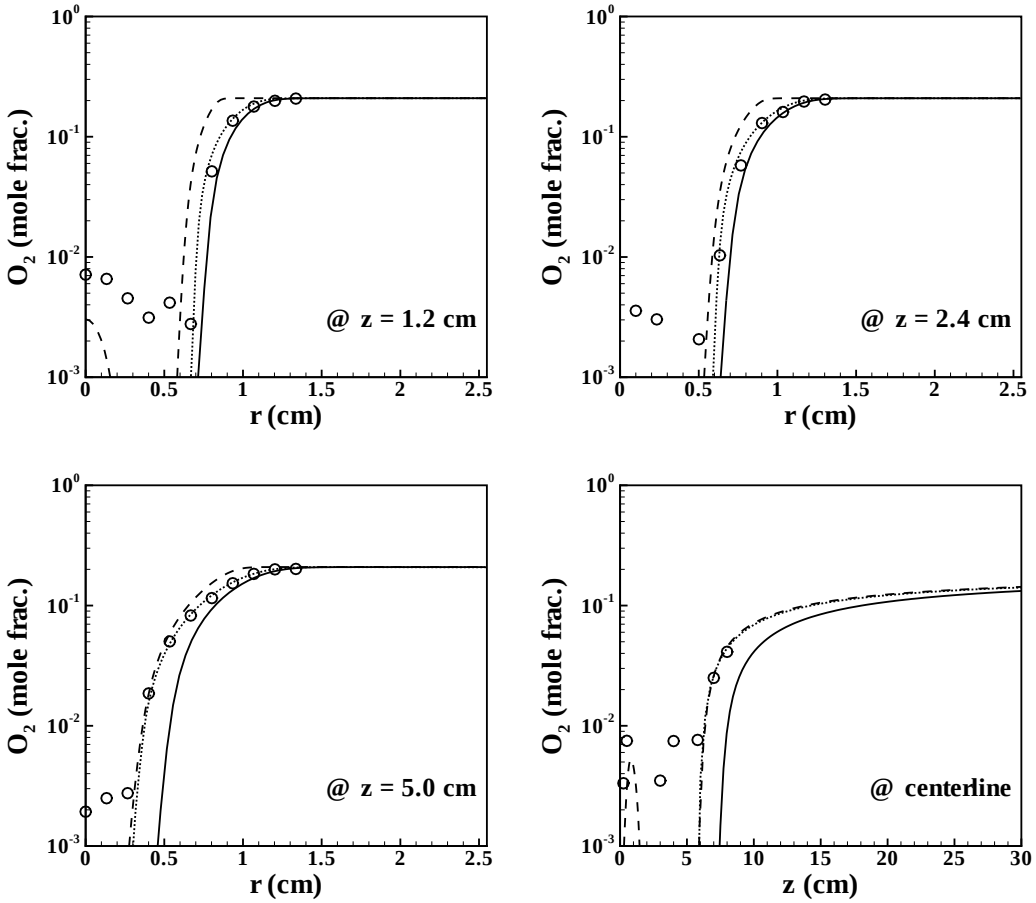


Figure 6.14: Radial and axial profiles of O_2 mole fractions at three axial locations and along the centerline.

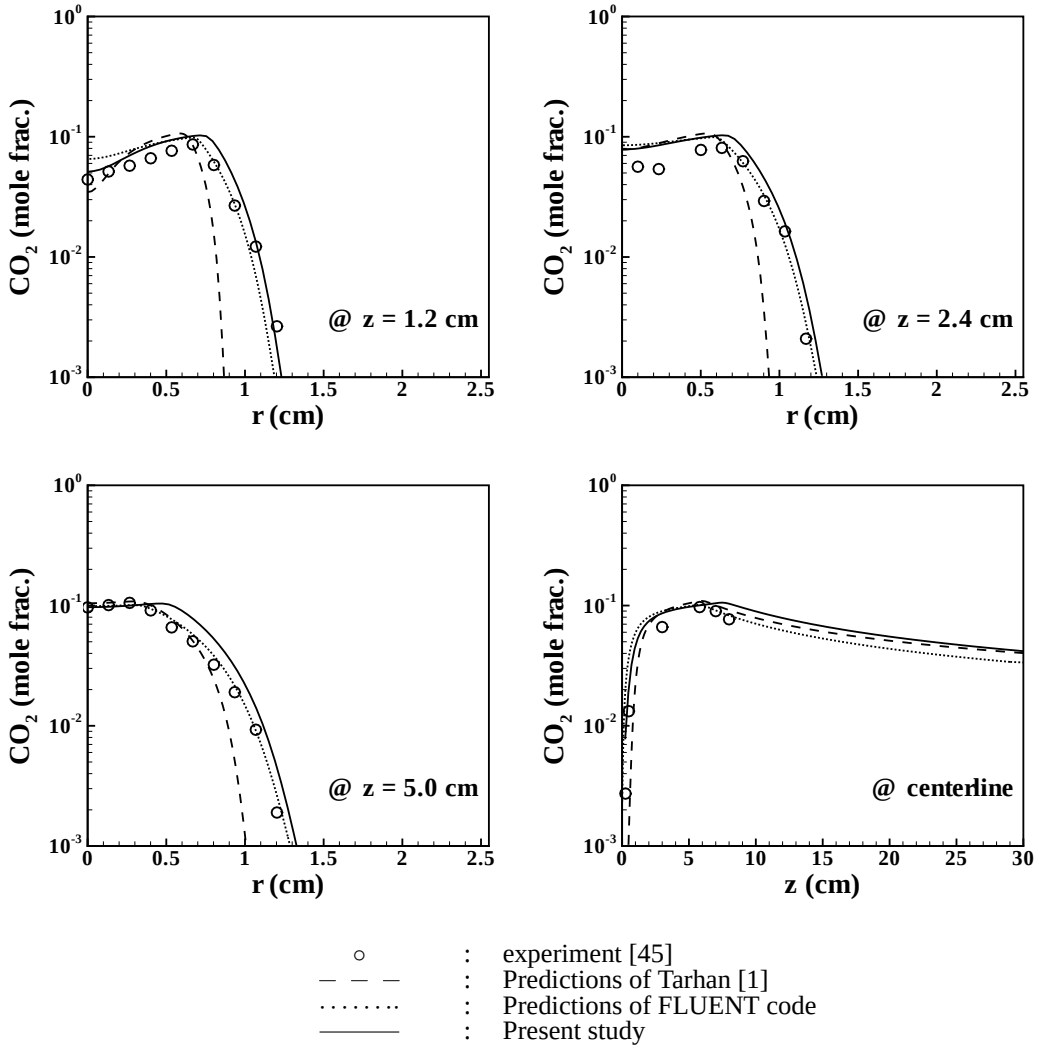


Figure 6.15: Radial and axial profiles of CO_2 mole fractions at three axial locations and along the centerline.

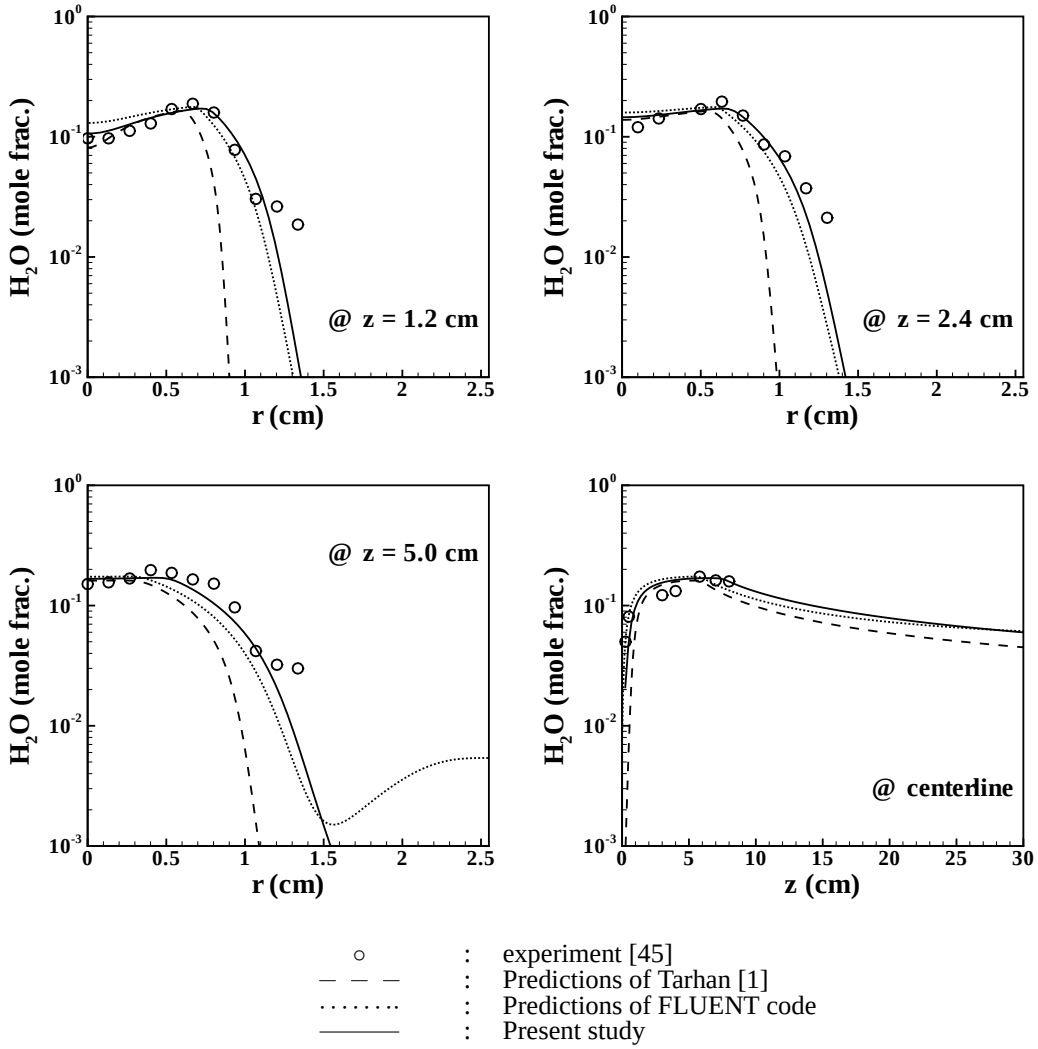


Figure 6.16: Radial and axial profiles of H_2O mole fractions at three axial locations and along the centerline.

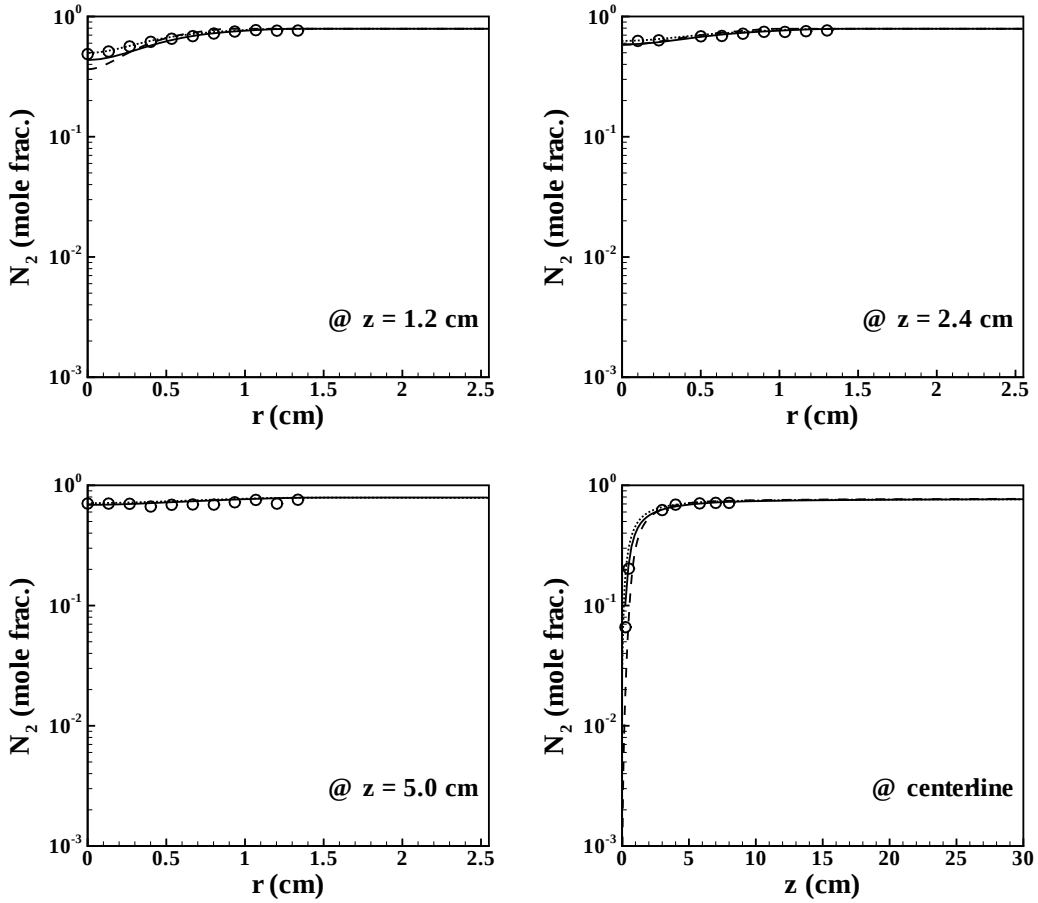


Figure 6.17: Radial and axial profiles of N_2 mole fractions at three axial locations and along the centerline.

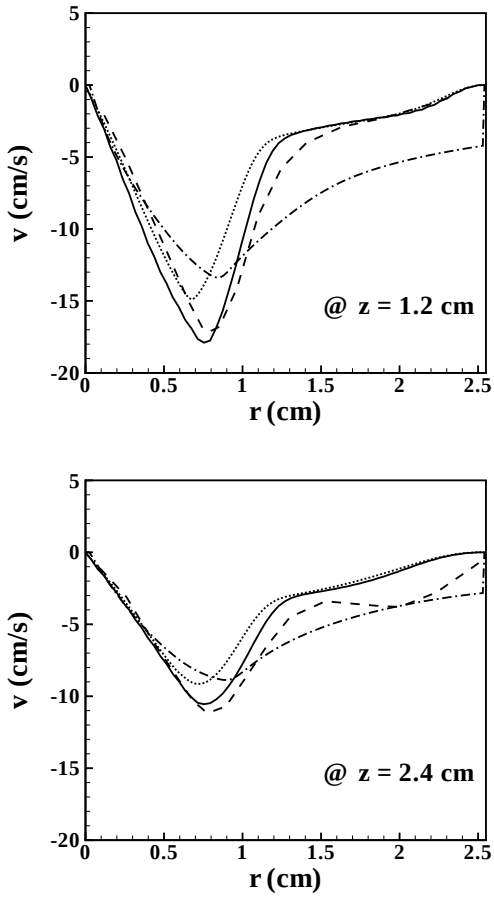


Figure 6.18: Radial profiles of radial velocity at two axial locations.

6.3.3 Comparison of Numerical Results With and Without Radiation

In order to investigate the effect of radiative heat transfer and the type of radiation model on the numerical results, the code was executed with gray and non-gray radiation models for the problem under consideration. The simulation with the gray radiation model was carried out in a transient fashion for a final time of t_f = 0.5 s. Time step used in the computations was 1×10^{-5} s and the number of grid points employed for the CFD and radiation codes were 65×129 and 9×17, respectively. Under these conditions, the simulation took 25 hours of CPU time. The execution with non-gray radiation model on the other hand, was performed by using the steady-state velocity, temperature and major species solutions of the computations without radiation as initial conditions and with the same time step and grid resolutions used for the simulation with gray radiation model. The CPU time required to obtain steady-state results for this case was found to be 158 hours.

The steady-state predictions obtained for the three cases: *i)* without radiation; *ii)* with gray radiation model; *iii)* with non-gray radiation model; are compared with the experimental data and other numerical solutions available in the literature. The radial profiles of temperature, axial velocity, mole fractions of CH_4, O_2, CO_2, H_2O and N_2 at three axial locations above the burner inlet and axial profiles along the centerline predicted by the code are plotted together with the experimental data [45, 46] and the numerical solutions of Uygur *et al.* [2] obtained with the previously developed code having parabolic pressure scheme, in Figures 6.19-6.25. The contour plots of the dependent variables for the three cases are also compared with the numerical solutions of Tarhan [1, 85] and Uygur *et al.* [2] in Figures 6.28-6.36. The relative importance of radiation with respect to conduction is demonstrated by plotting radiative and conductive heat fluxes at the tip of the flame. Finally, the efficiency of the code is discussed in terms CPU time of the executions.

The radial and axial profiles of temperature and axial velocity are displayed in Figures 6.19 and 6.20, respectively. As can be seen from the figures, no significant discrepancy can be noticed between the predictions with and without radiation at

the first axial location. However, the effect of incorporation of radiation model in the computations is visualized better as downstream locations are reached. The radial profiles with radiation at the second (z = 2.4 cm) and third (z = 5.0 cm) axial locations demonstrate that both axial velocity and temperature predictions follow the experimental trends closer when compared to the results without radiaton and numerical solutions of Uygur *et al.* [2]. In particular, the peak temperatures are in favorable agreement with the experimental data. The improvement of the results is physically consistent with the fact that radiative heat losses from the flame lower temperatures and in turn velocities. On the other hand, notwithstanding the decrease in temperatures, the flame height predictions remain unaffected and was found to be 7.4 cm for both radiation models as depicted in the axial temperature profile.

The major species profiles are illustrated in Figures 6.21-6.25. As can be seen from the figures, inclusion of radiation has only minor effect in species predictions. The profiles of CH_4 and O_2 agree reasonably well with the experimental data. This is attributed to the fact the effect of chemical reaction mechanism is more pronounced on the species predictions than the effect radiative heat transfer [2]. Nevertheless, the predictions of radiatively participating gases, CO_2 and H_2O, are in excellent agreement with the experimental measurements. This places further confidence in the accuracy of the radiative heat transfer calculations since the computation of absorption coefficients and hence the radiative source term with both gray and non-gray radiation models rely on the concentrations of CO_2 and H_2O.

In an attempt to demonstrate the relative significance of radiative heat transfer with respect to transfer by conduction, radial profiles of radiative and conductive heat fluxes in r-direction at the tip of the flame ($z \approx 7.4$ cm) are plotted in Figure (6.26). As can be seen from the figure, inside the flame, conduction is the superior mode of heat transfer owing to high temperature gradients occurring at this region whereas radiation is approximately ten times the transfer by conduction outside the flame. This behavior is consistent with the numerical results of Zhang [86] on the problem under consideration.

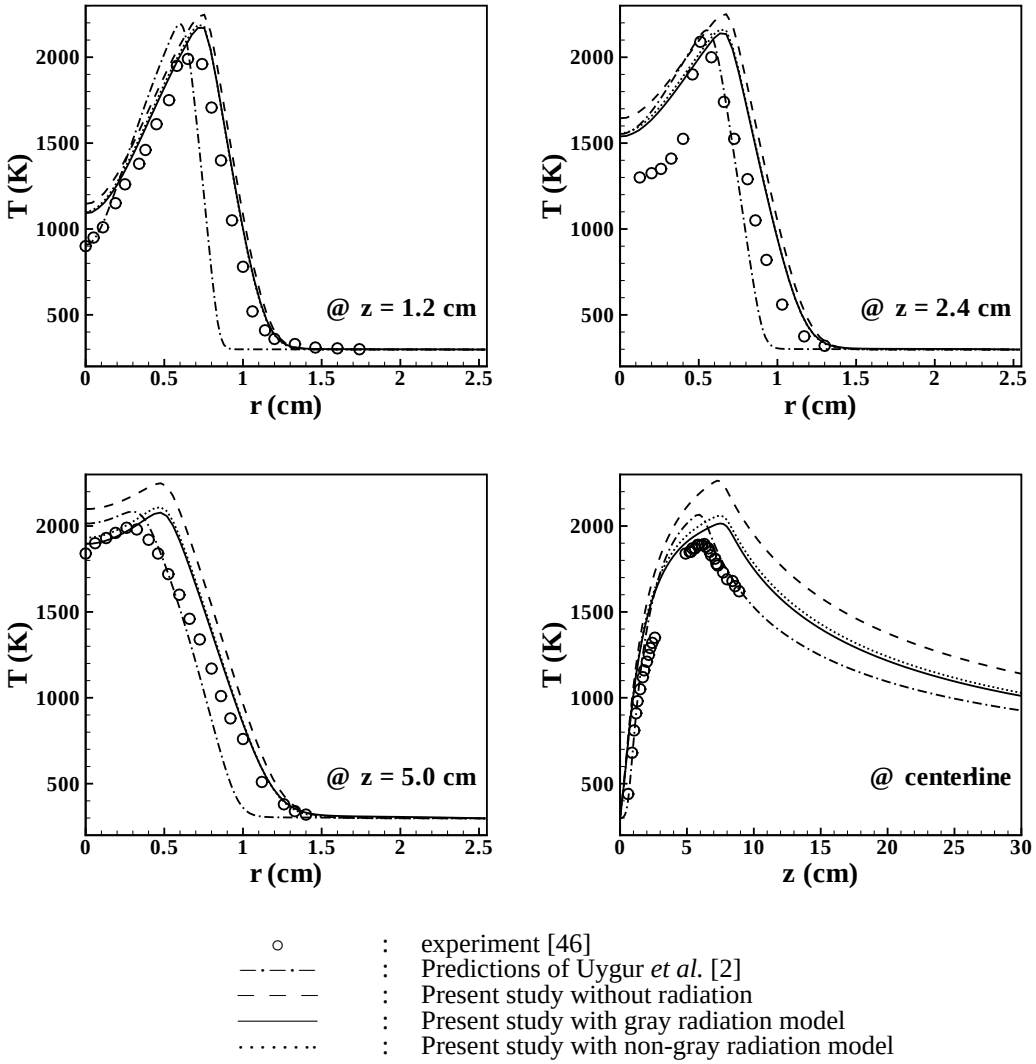


Figure 6.19: Radial and axial profiles of temperature at three axial locations and along the centerline.

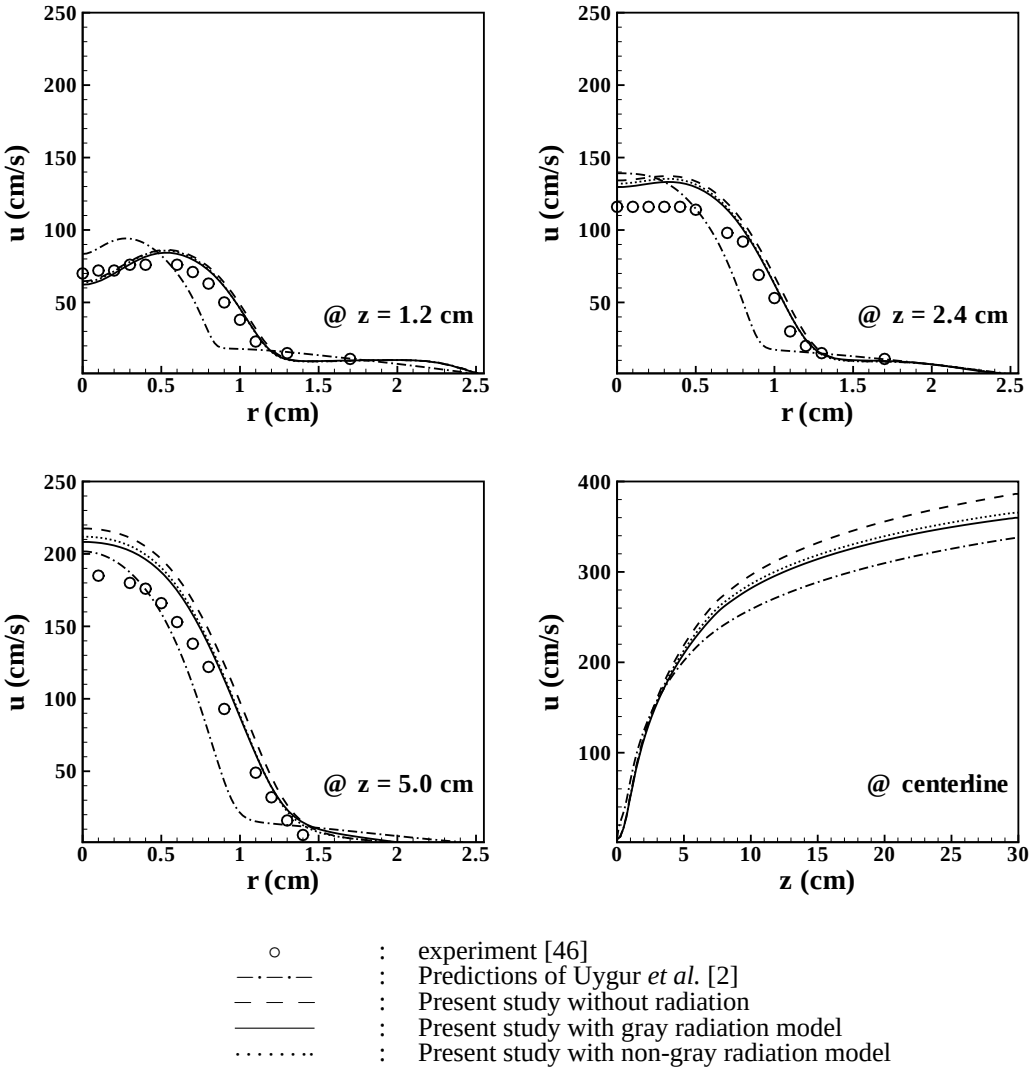


Figure 6.20: Radial and axial profiles of axial velocity at three axial locations and along the centerline.

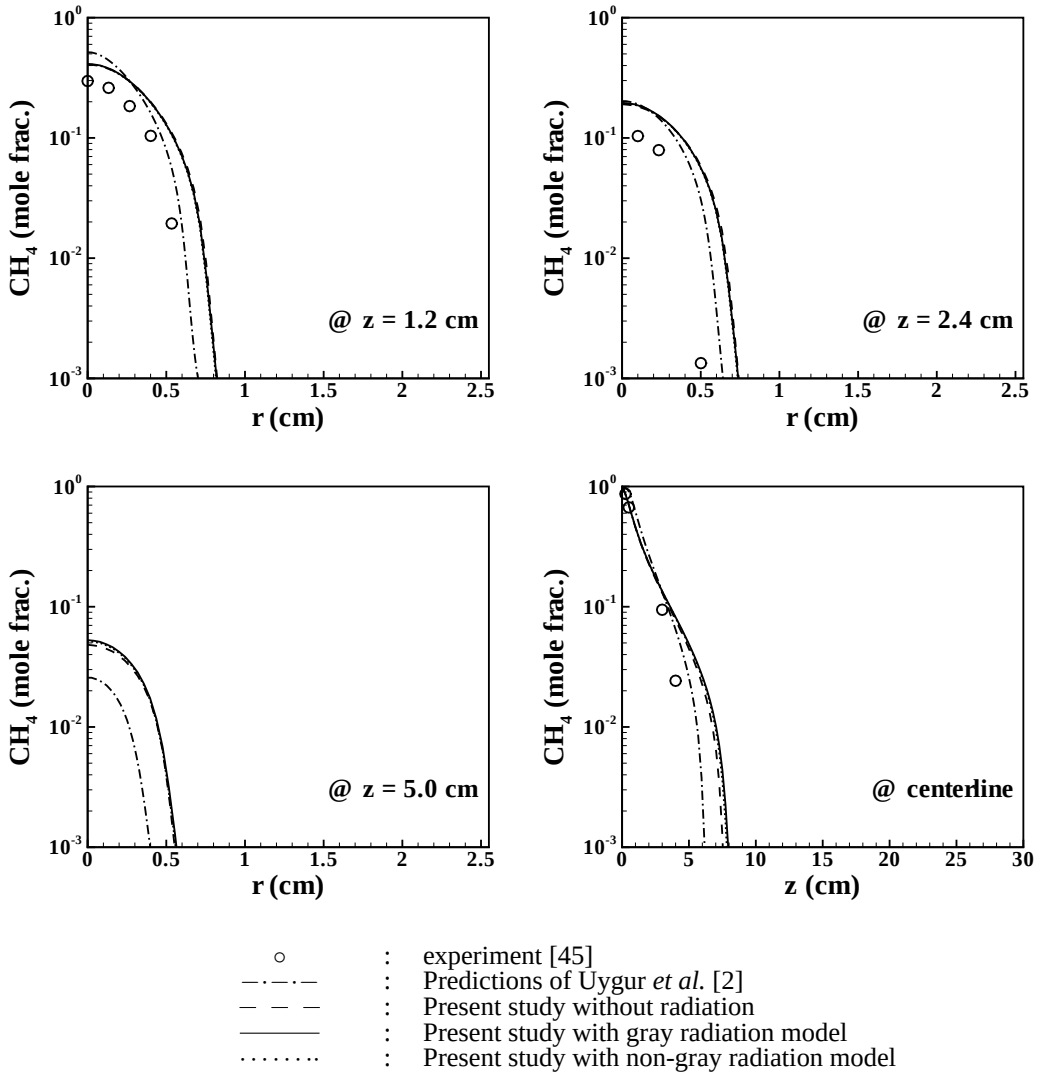


Figure 6.21: Radial and axial profiles CH_4 mole fractions at three axial locations and along the centerline.

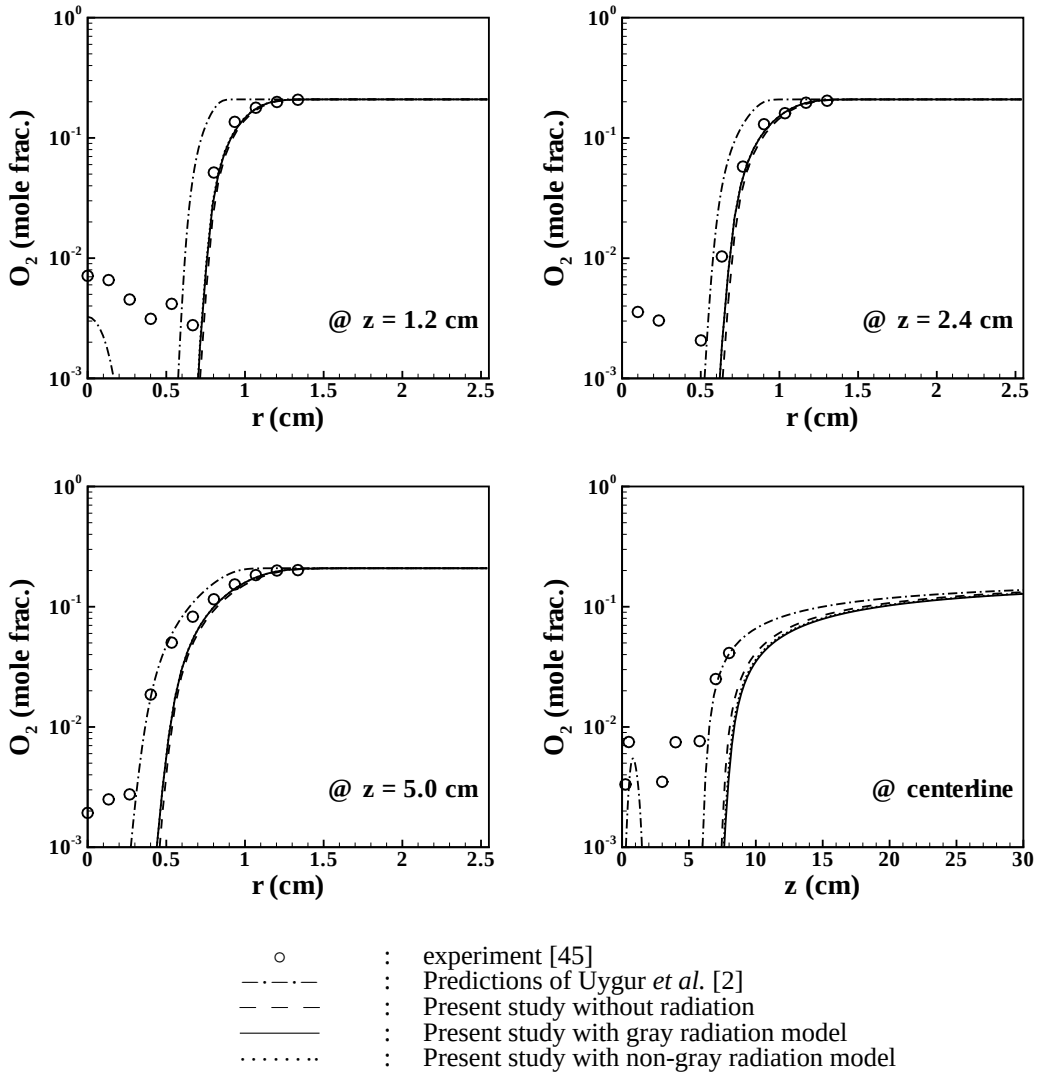


Figure 6.22: Radial and axial profiles O_2 mole fractions at three axial locations and along the centerline.

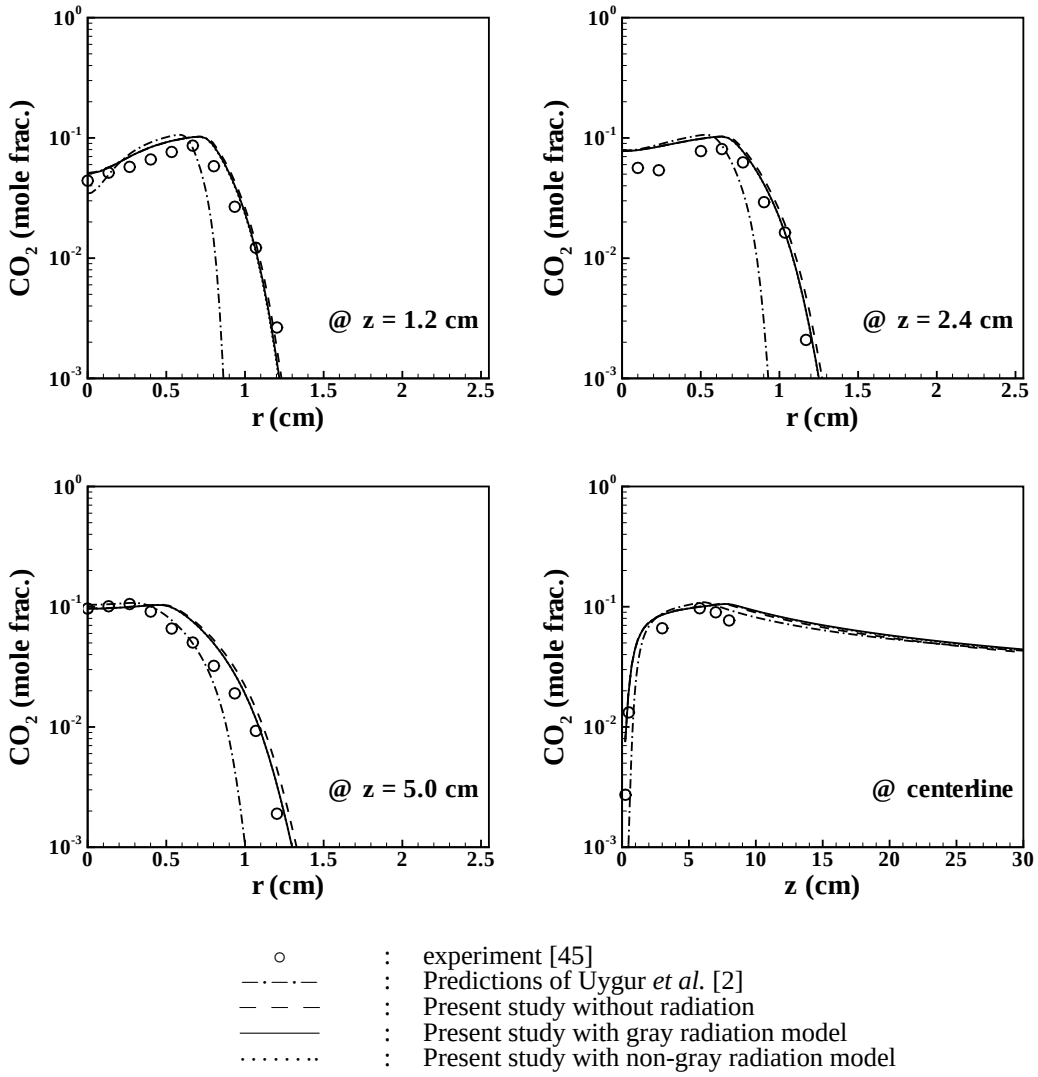


Figure 6.23: Radial and axial profiles CO_2 mole fractions at three axial locations and along the centerline.

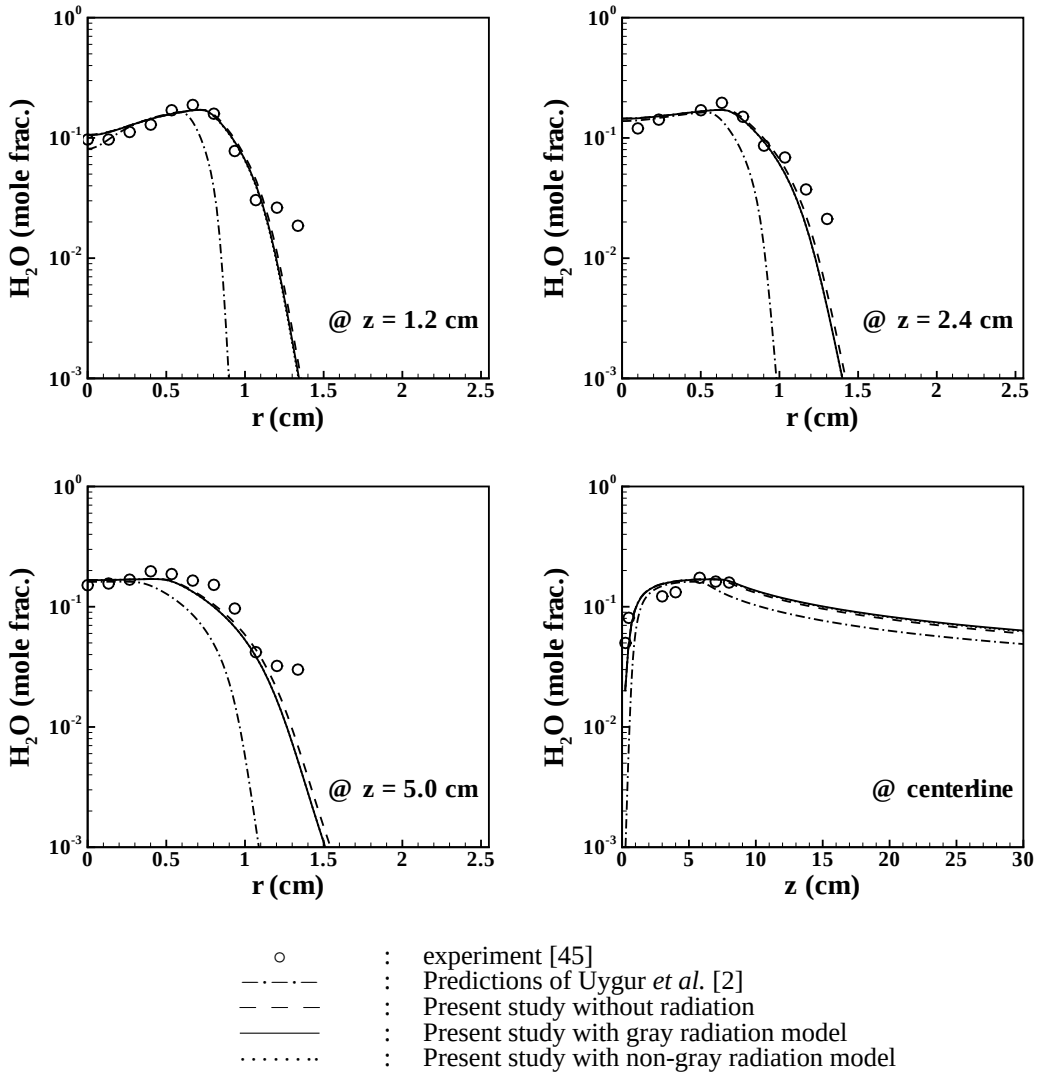


Figure 6.24: Radial and axial profiles H_2O mole fractions at three axial locations and along the centerline.

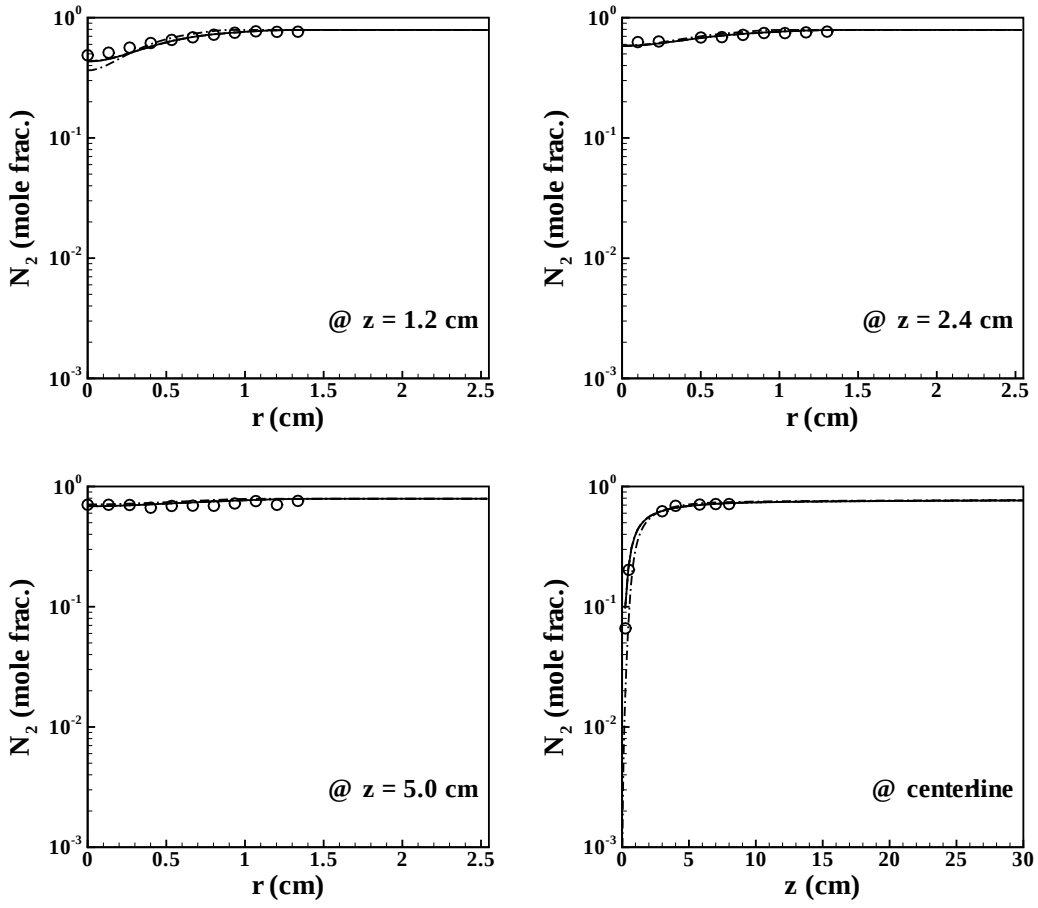


Figure 6.25: Radial and axial profiles N_2 mole fractions at three axial locations and along the centerline.

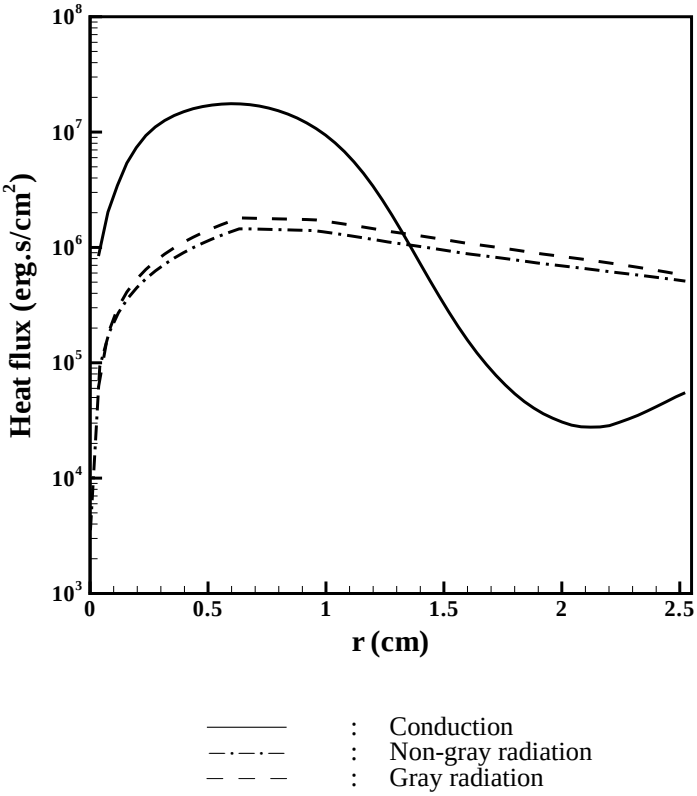


Figure 6.26: Conductive and radiative heat fluxes in *r*-direction at the tip of the flame.

In order to present the reader a complete picture of the axisymmetric diffusion flame under consideration and perform a comparison of the results with and without radiation based on the complete flow field, steady-state contours of axial and radial velocities, temperature, pressure, pressure and major species isopleths are illustrated in Figures 6.27-6.30.

As can be seen from the Figure 6.27, temperature fields exhibit the shape of a typical diffusion flame. As expected, incorporation of radiation yields lower flame temperatures as an outcome of the radiative heat losses. In particular, the difference between the temperature predictions with and without radiation is as high as 200 K inside the flame. Comparison of the fields given in the figure reveals that the flame predictions for all three cases presented in this study are thicker and longer than those obtained by Tarhan [1] (Figure 6.27(d)) and Uygur *et al.* [2] (Figure 6.27(e)).

Axial velocity contours display that the flow accelerates the in the core region. Negative velocities (flow reversal) occur in the vicinity of the wall as a consequence of the large buoyancy forces produced by the heat release from the chemical reaction. Inspection of the results with radiation (Figure 6.28(b)-(c)) shows that the effect of radiative loss presents itself as lowered velocities, specifically along the centerline.

The radial velocity contours given in Figure 6.29 show that the direction of the velocities near the burner inlet are towards the centerline which causes air to flow to the reaction zone. Also displayed by the figure is the fact that the effect of radiation is not pronounced on the radial velocity field where all contour plots for all three cases are almost identical. However, comparison of present predictions with the numerical solutions of Tarhan [1] (Figure 6.29(d)) and Uygur *et al.* [2] (Figure 6.29(e)) displays significant disparities. This is attributed to the fundamental differences between the pressure based algorithm used in the present study and the parabolic scheme employed in [1, 2].

The pressure contours for all three cases are illustrated in Figure 6.30. As can be seen from the figure, pressure exhibits a linear trend and no significant variation in r-direction can be observed. The pressure drop accross the burner was found to be

about 40 dyne/cm^3.

The isopleths of major species are illustrated in Figures 6.31-6.35. As can be seen from Figures 6.31-6.32, methane is depleted after a short distance from the burner inlet and almost no oxygen is present inside the flame due to the global reaction mechanism employed in the computations. Since the majority of the production of CO_2 and H_2O takes place at high temperatures, their isopleths exhibit the same trend with the temperature contours (Figure 6.33-6.34). Nitrogen is the excess species and is present everywhere in the system except the burner inlet (Figure 6.35). An overall examination of the figures reveals that incorporation of radiation has a minor effect in the prediction of species.

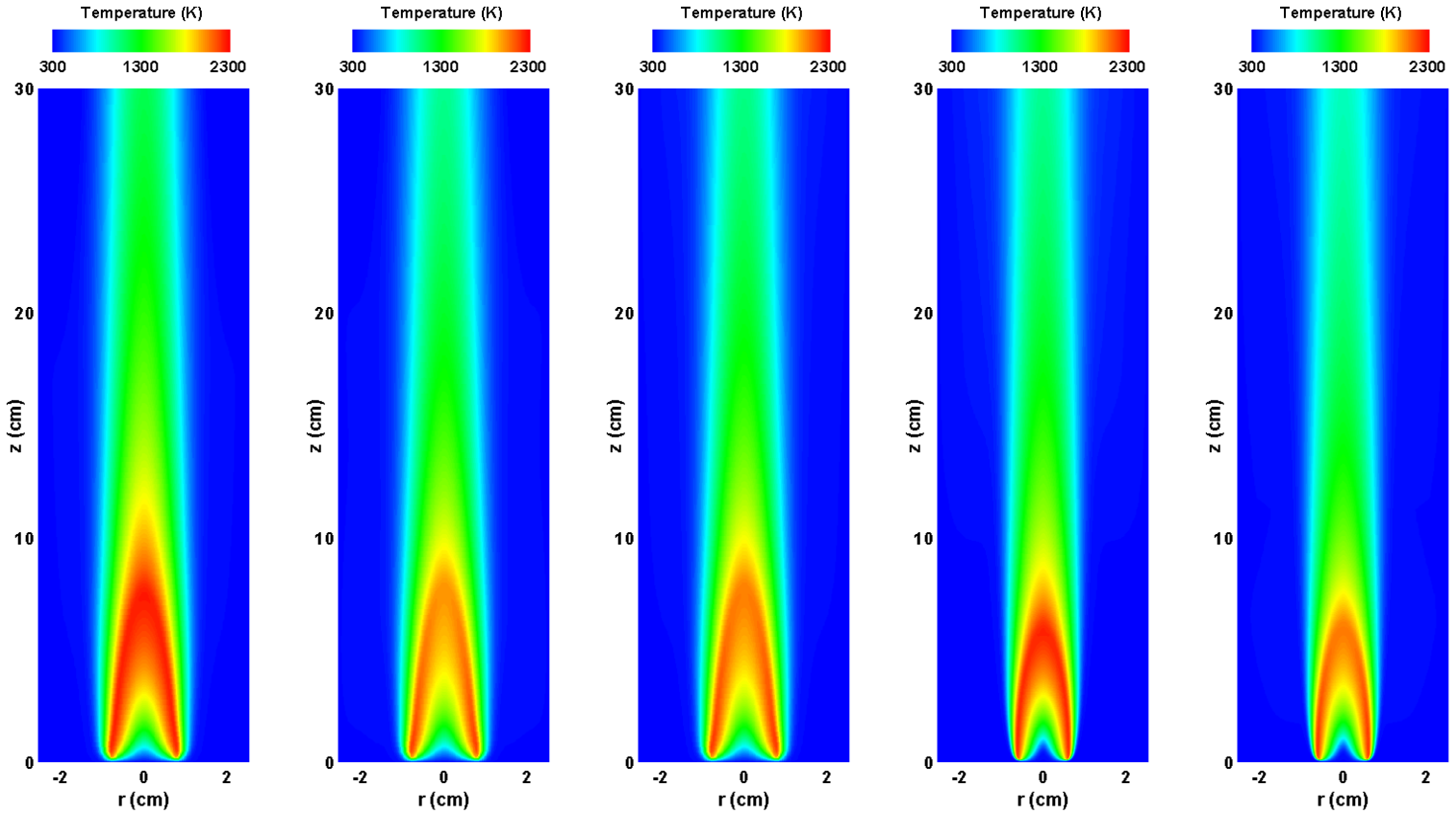


Figure 6.27: Comparison of temperature contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

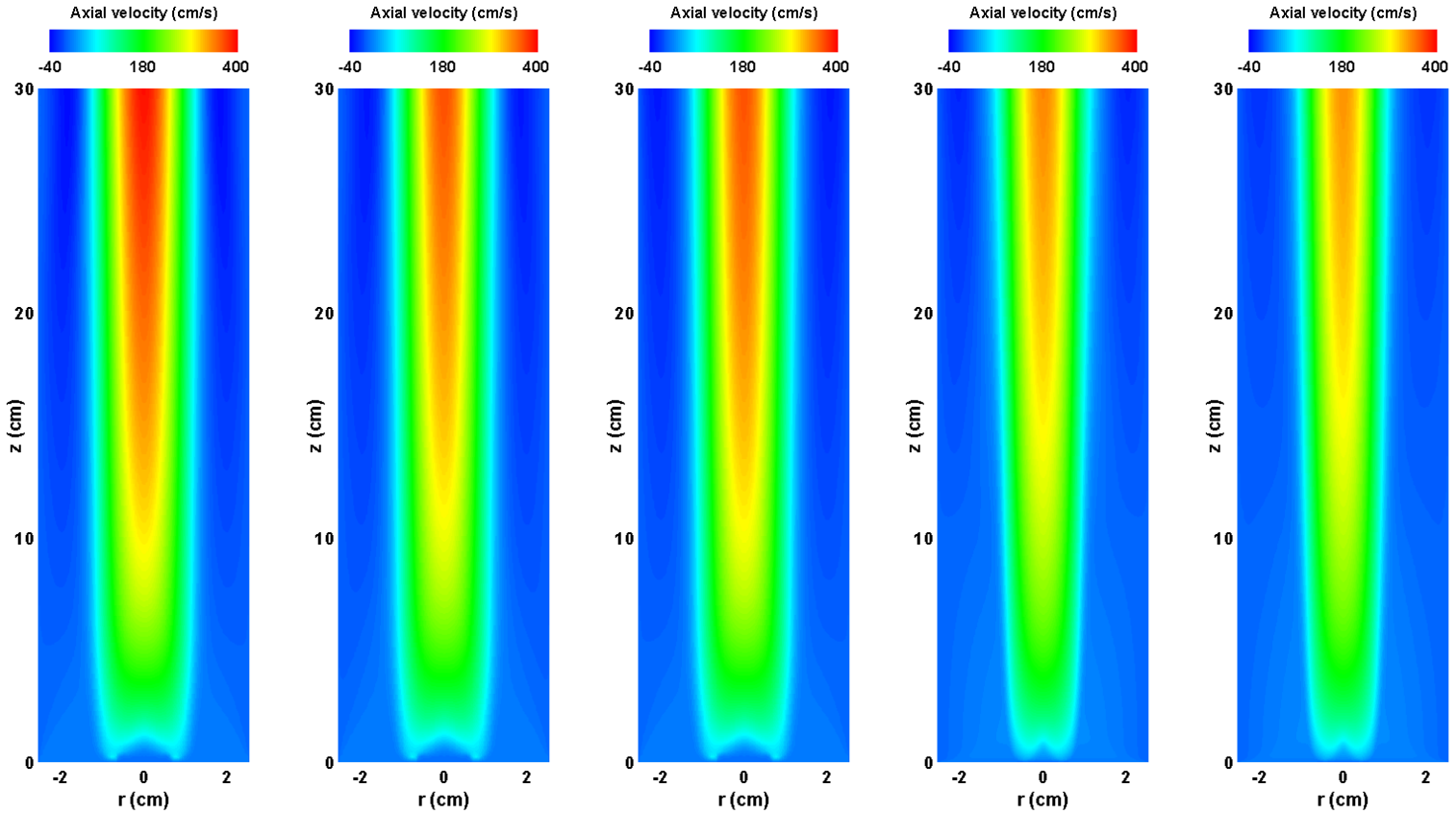


Figure 6.28: Comparison of axial velocity contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

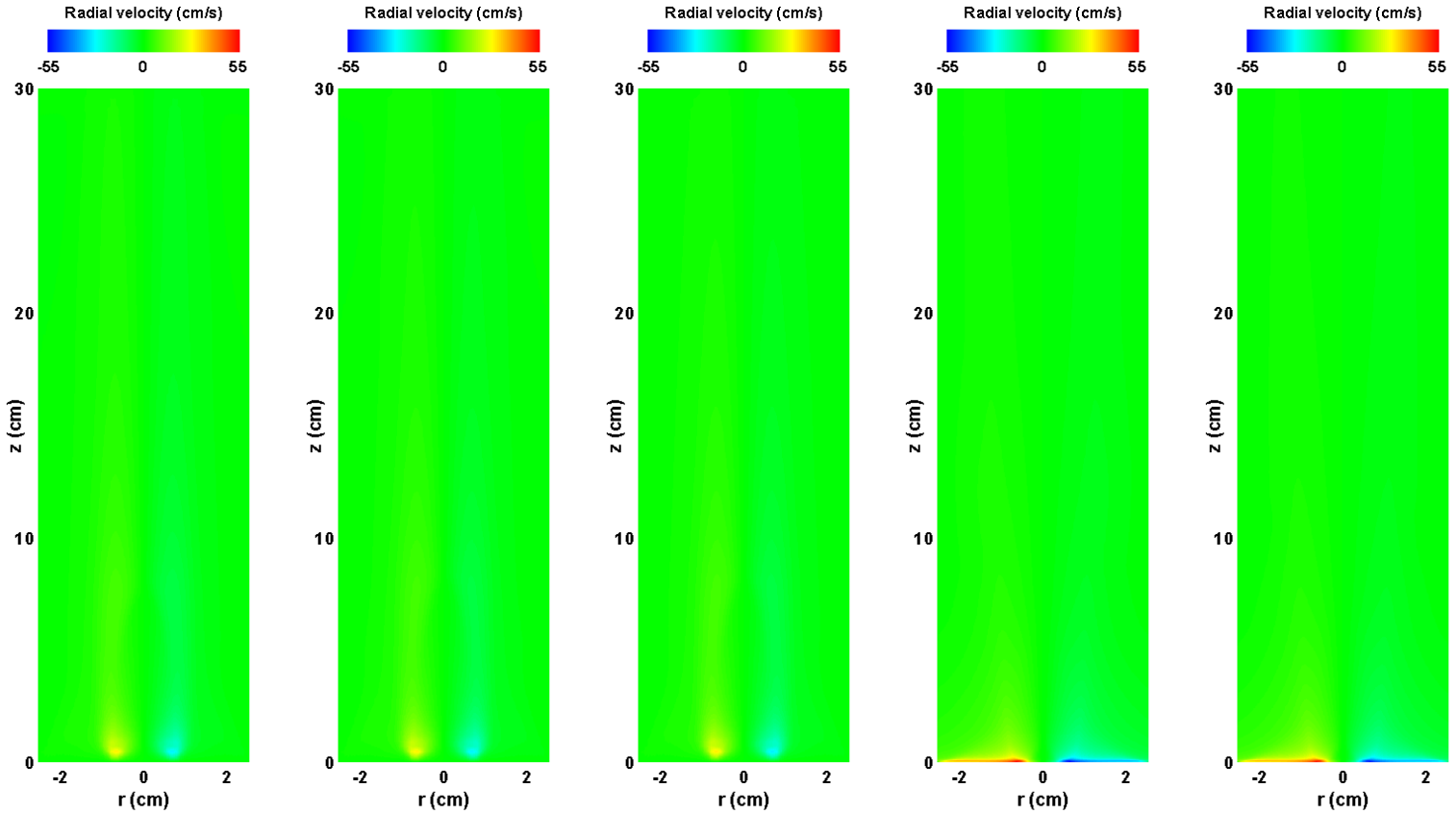


Figure 6.29: Comparison of radial velocity contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

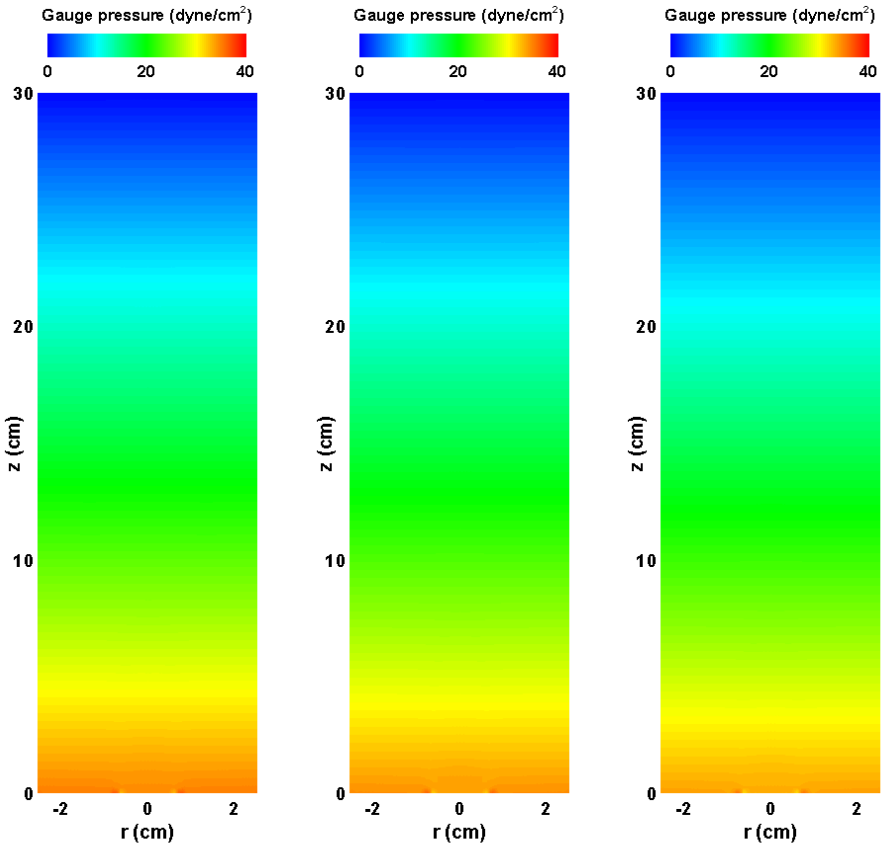


Figure 6.30: Comparison of pressure contours: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model.

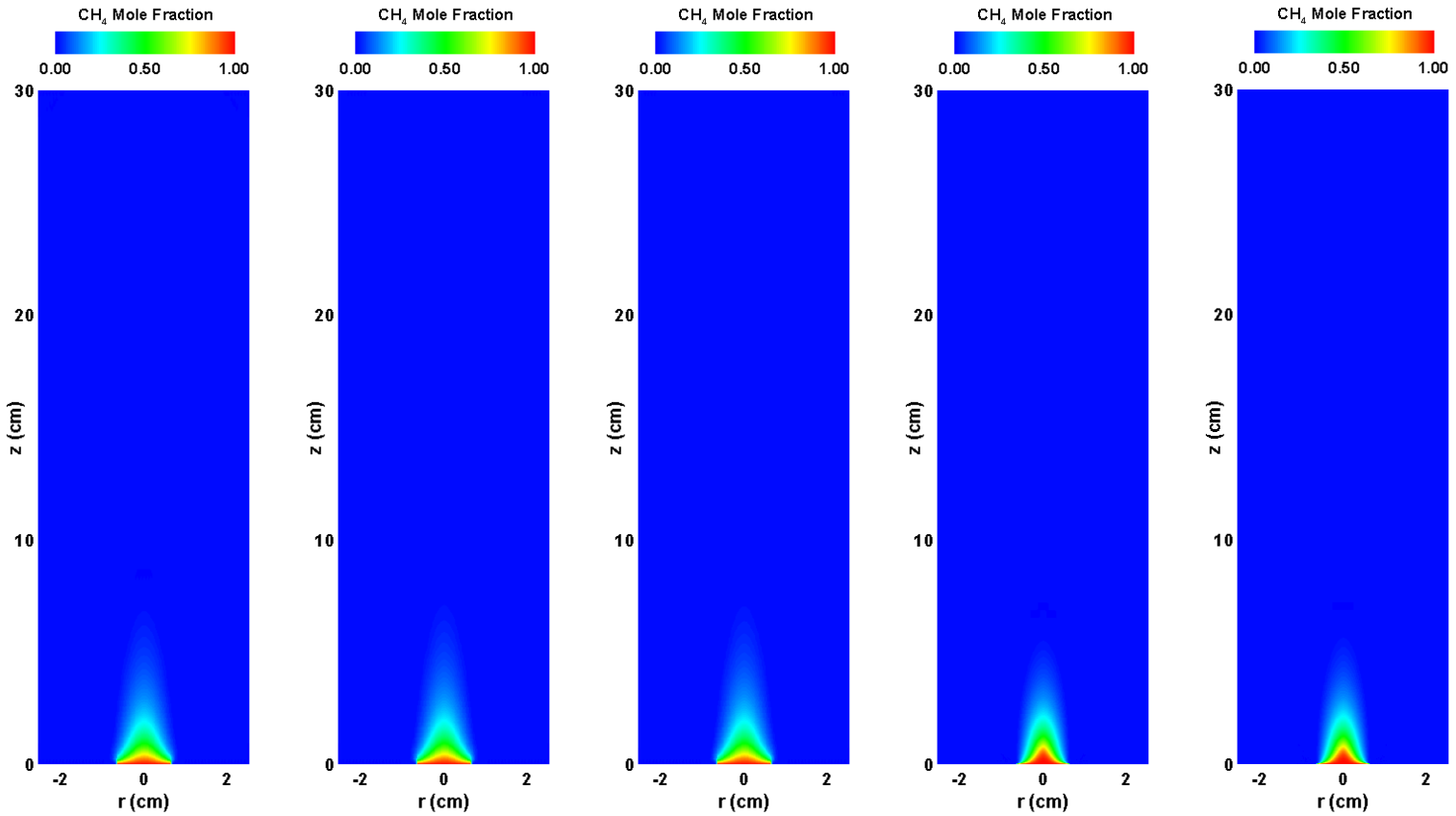


Figure 6.31: Comparison of CH_4 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

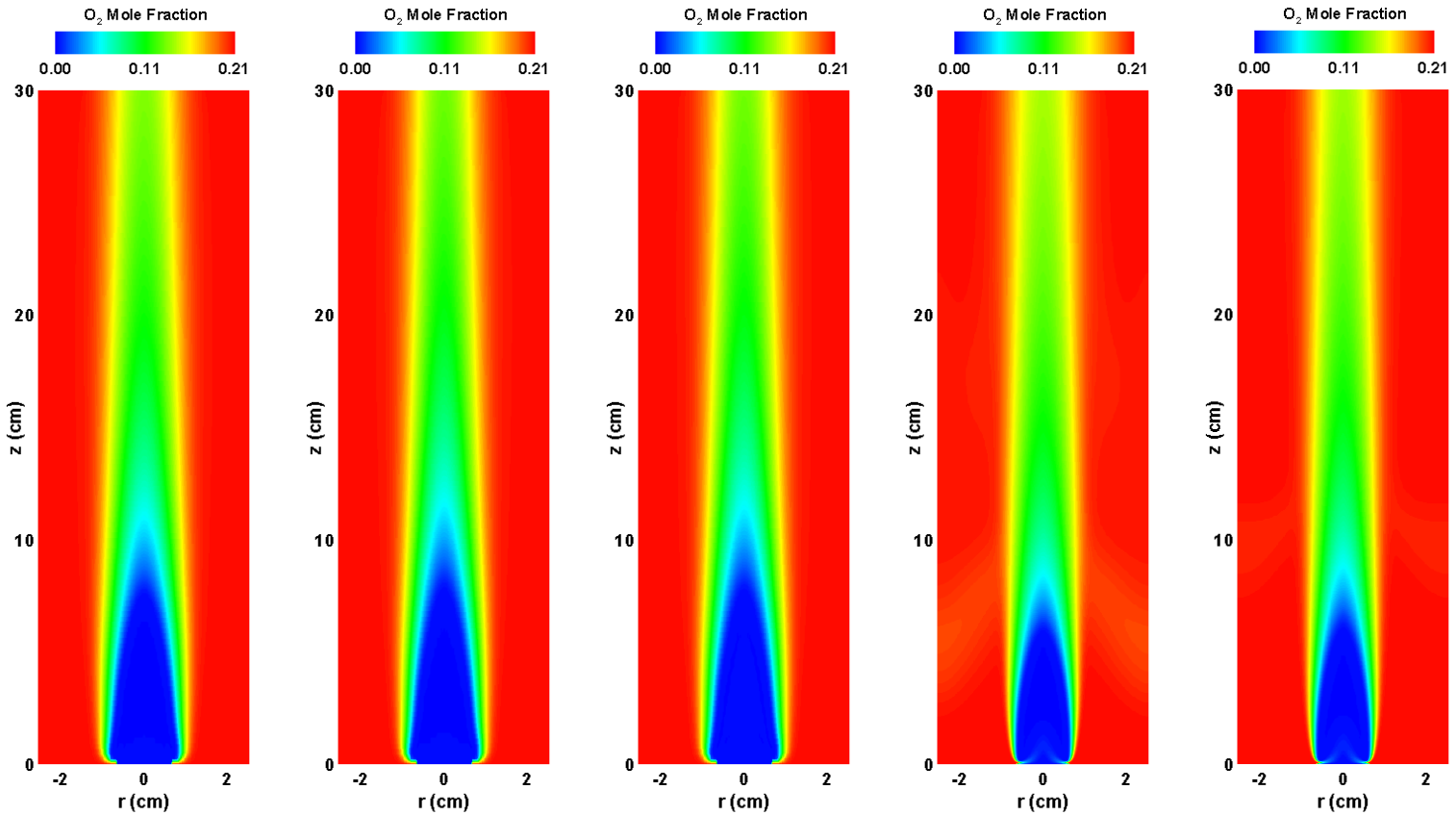


Figure 6.32: Comparison of O_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

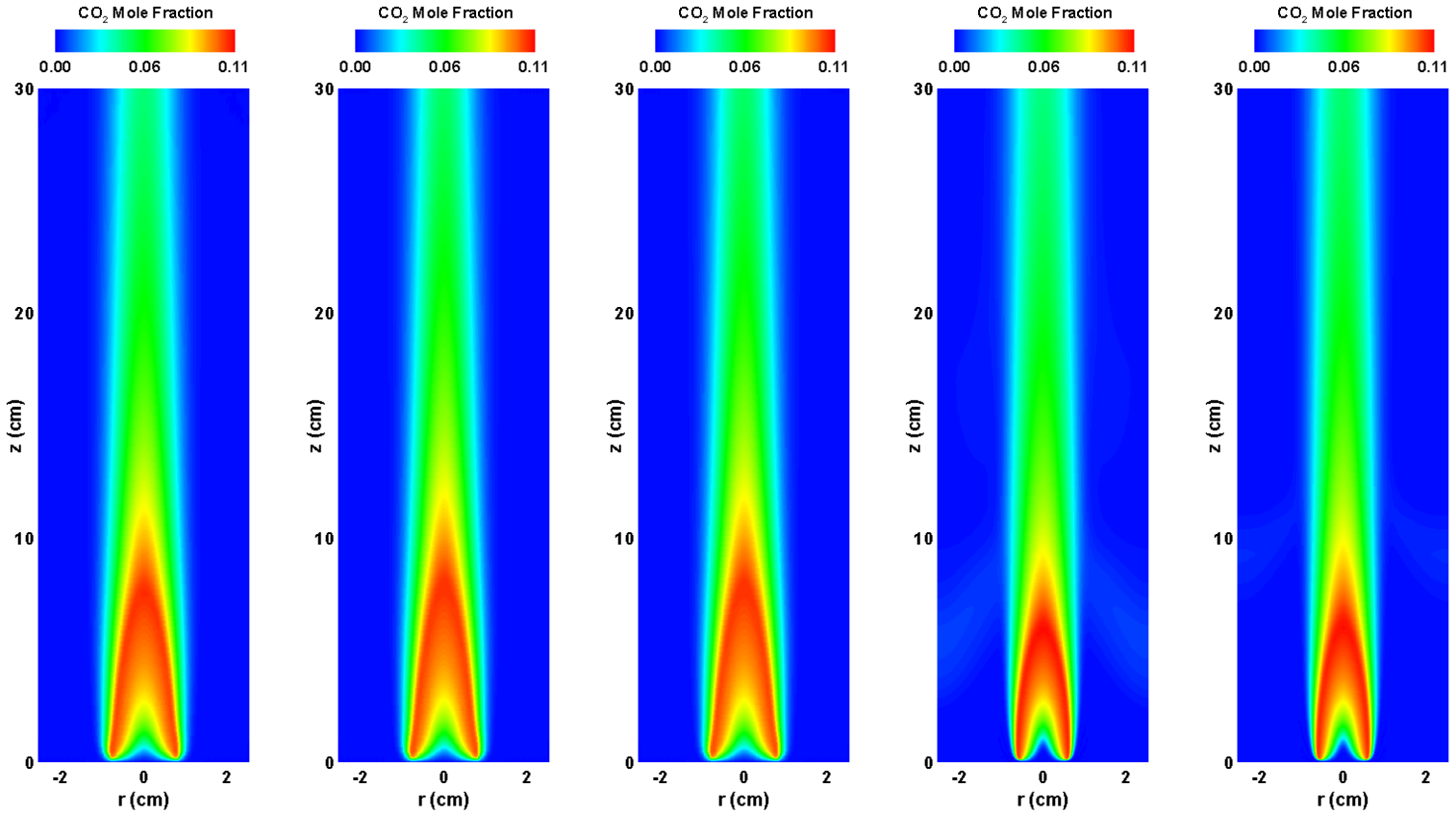


Figure 6.33: Comparison of CO_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

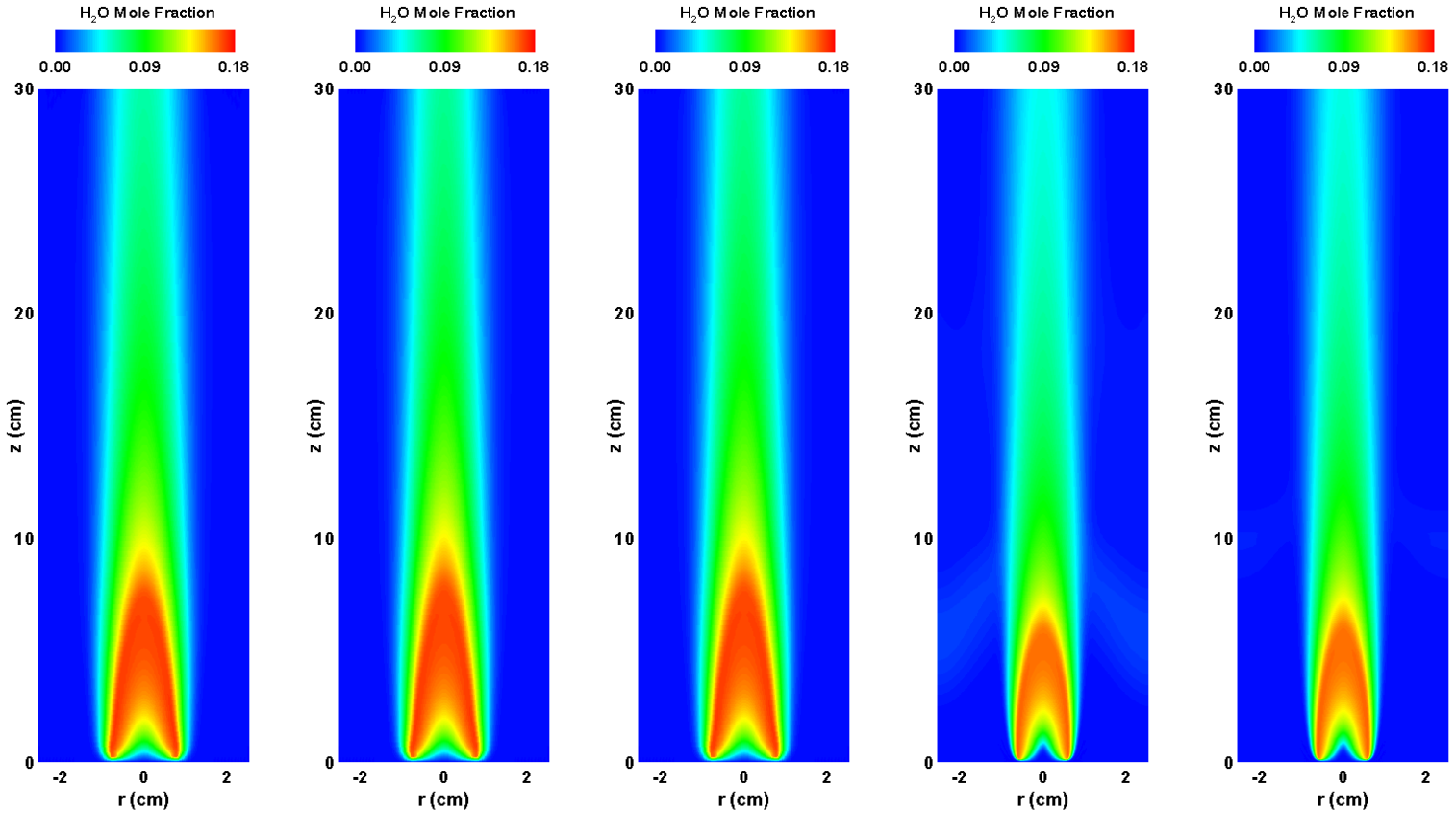


Figure 6.34: Comparison of H_2O isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

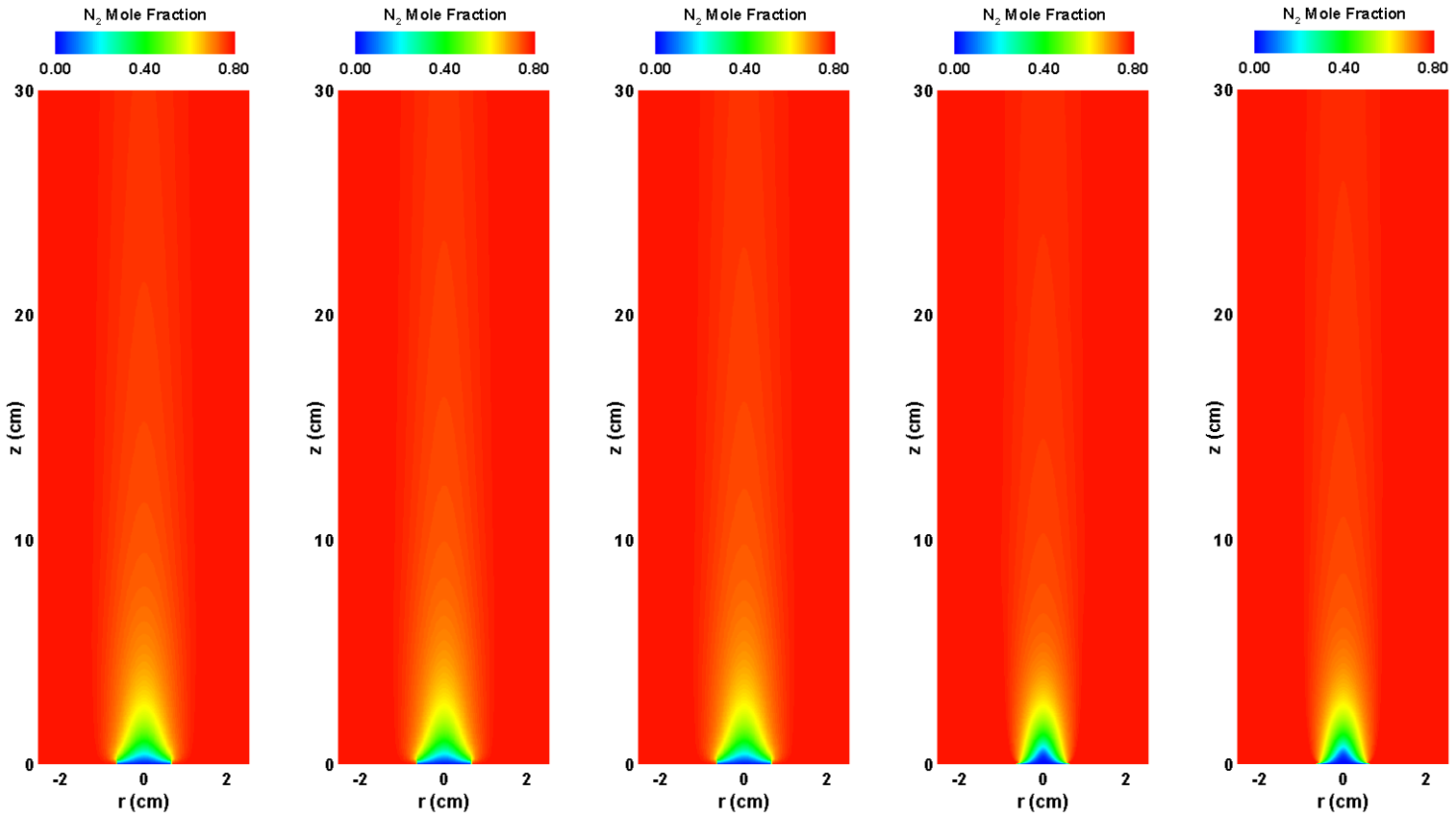
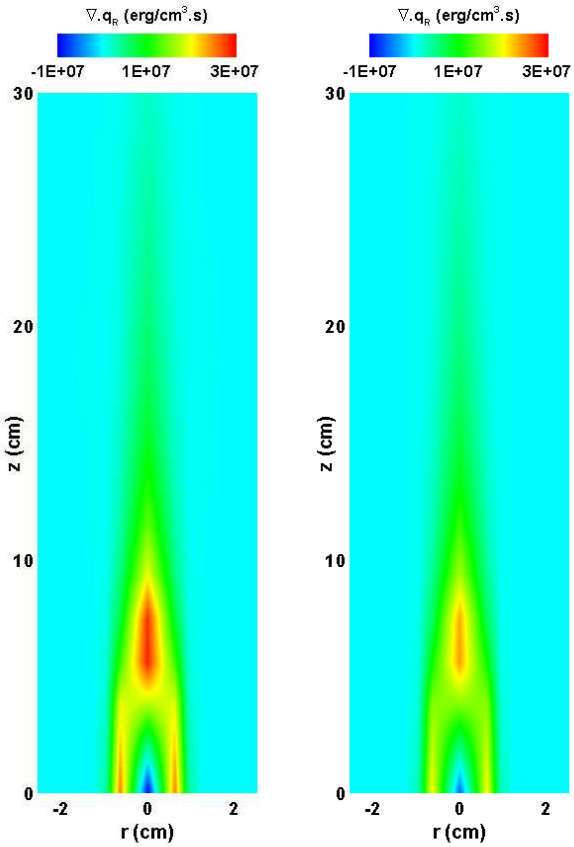


Figure 6.35: Comparison of N_2 isopleths: (a) Present study without radiation; (b) Present study with gray radiation model; (c) Present study with non-gray radiation model; (d) Predictions of Tarhan [1] (without radiation); (e) Predictions of Uygur *et al.* [2] (with gray radiation).

6.3.4 Comparison of Numerical Results With Gray and Non-gray Radiation

The comparison of the divergence of radiative heat flux fields obtained with gray and non-gray radiation models are illustrated in Figure 6.36. Figure shows that the two plots exhibit similar trends but with different magnitudes. Differences between the two fields can reach up to 100 % around the centerline. However, inspection of the results presented so far surprisingly reveals that the effect of these discrepancies on the predictions obtained with gray and non-gray radiation models is insignificant. It is true that in some regions of the flame the differences in the radiative source terms correspond to temperature differences as high as 80 K. Yet its effect on the major species predictions remain limited. Moreover, the computational cost brought by the non-gray radiation model [50, 60] employed in the present study is extremely high. Analysis shows that CPU time requirement of the executions with non-gray model is approximately 30 times more than that with the gray radiation model.



(a) (b)

Figure 6.36: Comparison of divergence of heat flux contours: (a) Present study with gray radiation model; (b) Present study with non-gray radiation model.

6.3.5 Transient Results

In order to demonstrate the predictive ability of the algorithm developed in the present study for transient solutions, time development of velocity and temperature fields obtained by executing the code with gray radiation model for the suddenly started diffusion flame are presented in Figs. (6.37)-(6.38). The burner is initially filled with air at room temperature and the flow is at rest. Fuel and air both at room temperature are allowed to enter the system and ignition takes place at the intersection region of fuel and air by providing a small hot region (which is at 1500 K) higher than the ignition treshold value for a time period of 50 ms. Combustion starts immediately and flame propagates to the burner exit.

Figure 6.7 shows the time development of the axial velocity by color contours and streamlines. As can be seen from the figure, as soon as flow is started, the velocity increases in the inlet region along the centerline due to increase in temperature. The flow starts to separate downstream yielding two large recirculation cells that are established on each side of the hot flame region. At steady-state lower cells merge with the upper ones, taking their final forms, and remain in the system. Air is entrained into the system at the system outlet to balance the momentum of the inlet fuel and air streams along with the frictional losses at the shield wall. The presence of these recirculation cells reduces the total area available for the flow of the combustion gases and hence the velocities are increased due to the combined effects of natural convection and a reduced flow area. High velocity gradients appear in the whole domain, where axial velocity varies in the range of -30 cm/s to 360 cm/s.

Time development of temperature field is shown in Fig. 6.38. As can be seen from the figure, as soon as flow is started, reaction starts to take place immediately and high temperature region extends from the boundary of the fuel and oxidizer jets to the symmetry axis. As time progresses, fuel and oxidizer flow downstream resulting in the shape of a typical diffusion flame during which a bottle-neck type structure is formed due to the presence of flow recirculations. After a distance from the inlet, fuel is depleted and therefore, temperature starts to decrease further downstream.

In an attempt to compare the transient predictions of the present code with those obtained by Uygur *et al.* [2] (see Figures D.1-D.2) it was seen that the development phases of the flames exhibit significant discrepancies, particularly in terms of flame thicknesses where a broader flame was predicted with the present algorithm compared to one obtained by [2]. Morover, the final time required to obtain steady-state results was reported as 8.0 s as opposed to the 0.5 s used in the present study. This is again attributed to the fundamental differences between the pressure based algorithm used in the present study and the parabolic scheme employed in [2]. For comparison purposes, the transient solutions obtained by Uygur *et al.* [2] are presented in Appendix D.

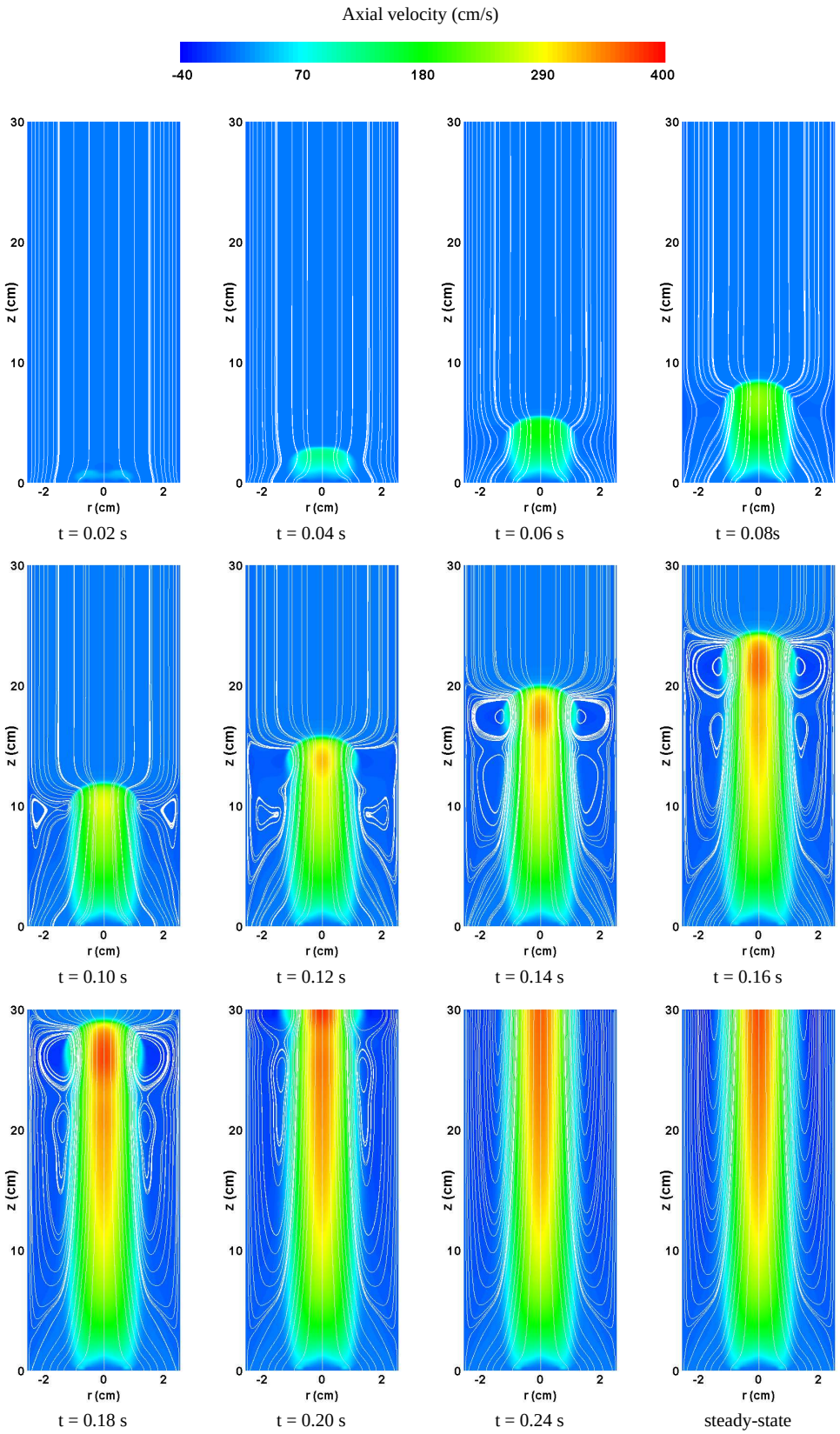


Figure 6.37: Time development of streamline pattern and axial velocity obtained with gray radiation model.

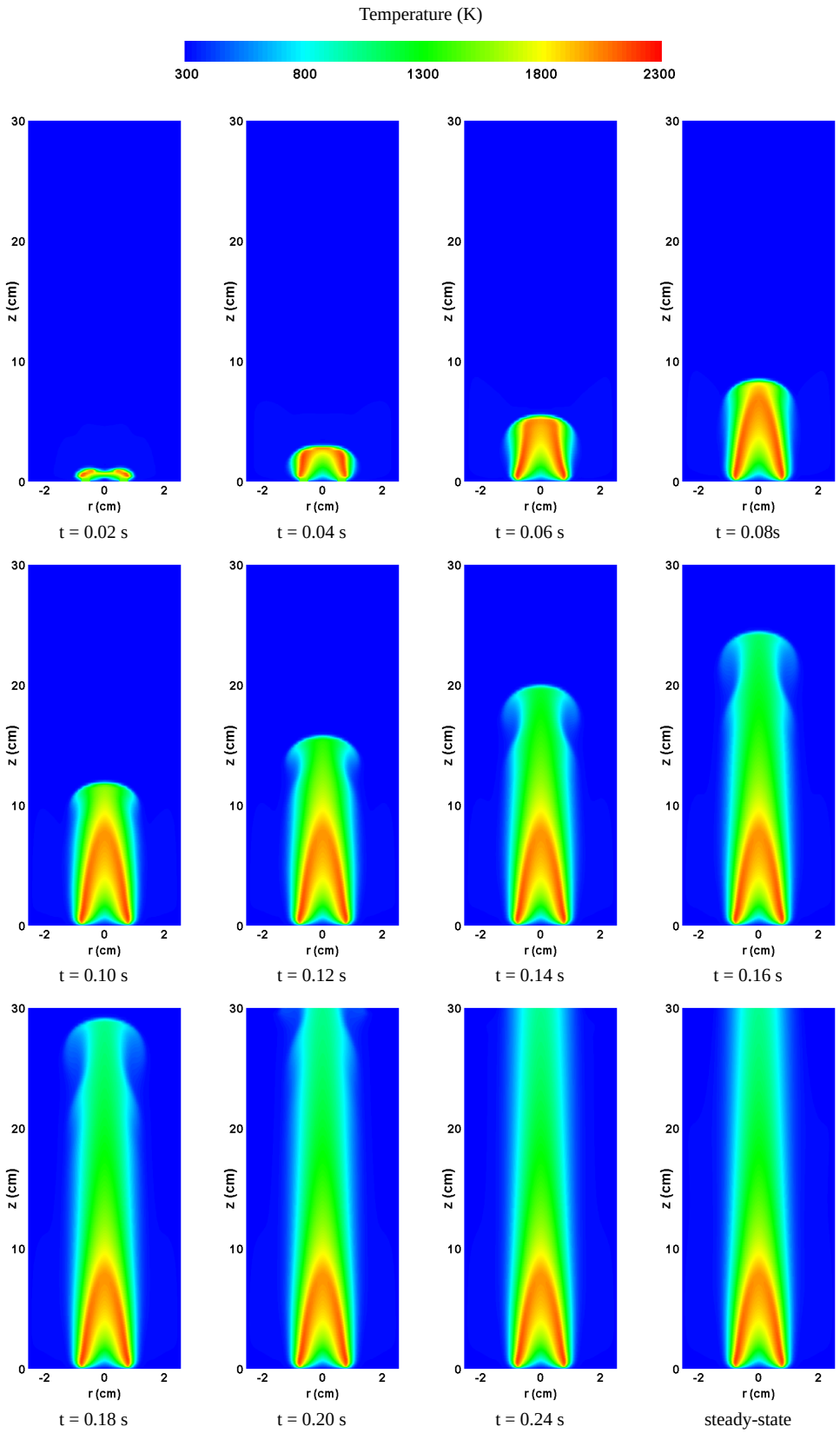


Figure 6.38: Time development of temperature field obtained with gray radiation model.

CHAPTER 7

CONCLUSIONS

In what preceeded, a non-iterative pressure based algorithm for the computation of transient reacting radiating flows was presented. The algorithm consists of splitting the solution of momentum, energy and species equations into a sequence of predictor corrector stages. A semi-discrete approach called the Method of Lines (MOL) which enables implicit time-integration at all splitting stages was used for the solution of conservation equations. The solution of elliptic pressure equation for the determination of pressure field was performed by a multi-grid (MUDPACK package) solver. Radiative heat transfer calculations were carried out by means of incorporating previously developed gray and non-gray radiation models into the algorithm. A first order (global) reaction mechanism was employed to account for the chemistry.

The predictive performance of the algorithm was first demonstrated on a non-isothermal flow problem involving turbulent flow of air in a strongly heated pipe. The steady-state predictions of the code were compared against the experimental data, DNS results and numerical solutions obtained with the previously developed code. Time development of axial velocity and temperature fields were illustrated.

The code was then applied to the prediction of laminar methane-air diffusion flame problem. The steady-state predictions obtained for the following cases: *i)* without radiation; *ii)* with gray radiation model; *iii)* with non-gray radiation model; are validated against the experimental data and other numerical solutions available in the literature. The effect of radiation and non-gray treatment of the radiative properties

on the numerical results were investigated. The capability of the code to predict transient reacting radiating flows was demonstrated on the same problem.

The following is a summary of the conclusions drawn at the end of the present investigation:

- The MOL is an accurate and reliable method for the solution of conservation equations enabling implicit time-integration at all splitting stages without extra complexity in the formulations.
- The multi-grid solver employed in the code is an efficient tool for the solution of elliptic pressure equation which makes time-dependent computations feasible.
- Execution of the algorithm with more than one corrector stage do not bring additional accuracy and increases the computational effort significantly.
- The predicted velocity and temperature fields for non-isothermal pipe flow problem compare favorably with the experimental data and DNS solutions. The transient solutions display expected trends.
- Despite the simplicity of the reaction mechanism employed in the computations, steady state velocity, temperature and major species mole fraction predictions obtained with and without radiation for the laminar methane-air diffusion flame problem are overall in good agreement with the experimental data.
- Incorporation of radiation transport in the simulations has significant effect on the resulting velocity and temperature fields and improves the predictions considerably. Yet its effect is found to be minor on major species predictions.
- Executions with both radiation models reveal that the non-gray radiation model considered in the present study produces similar results with the gray model at a considerably higher computational cost.

On the whole, the algorithm developed is proved to be an efficient and versatile tool for the time-dependent computation of different flows scenarios and its extension to turbulent reacting radiating flows with the improvement of the existing models is highly promising.

7.1 Suggestions for Future Work

Based on the experience gained in the field of CFD, the followings are suggested for the future extension of the work.

- The predictive accuracy of the code can be tested with multi-step reaction mechanisms involving minor species formation such as CO and NO.
- The results presented in this study were obtained by employing the same ODE solver (LSODES) for the solution of all conservation equations. However, the algorithm allows the use of different ODE solvers at each splitting phase. The code can be further improved in terms of CPU efficiency by the utilization of different ODE solvers with varying stiffness for the solution of each conservation equation.
- Simple and complex soot models can be incorporated to the code to investigate the soot formation process and its interaction with radiative heat transfer.
- For an accurate and proper representation of turbulent flows, the formulation used in the code should be extended to three dimensions.
- Considering the high computational cost of simulating turbulent flows due to fine grid resolution requirements, the algorithm should be parallelized.

REFERENCES

[1] T. Tarhan, “Numerical simulation of laminar reacting flows”, *PhD Thesis*, Middle East Technical University, Turkey, 2004.

[2] A. B. Uygur, T. Tarhan, and N. Selçuk, “Transient simulation of reacting radiating flows”, *Int. J. Therm. Sciences*, Vol. 45, No. 10, 969–976, 2006.

[3] F. H. Harlow and J. E. Welch, “Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface”, *Phys. Fluids*, Vol. 8, No. 12, 2182–2189, 1965.

[4] S. V. Patankar and D. B. Spalding, “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”, *Int. J. Heat Mass Transfer*, Vol. 75, 3787–1806, 1972.

[5] S. V. Patankar, *Numerical heat transfer and fluid flow*. United States: Hemisphere, 1980.

[6] J. P. VanDoormaal and G. D. Raithby, “Enhancements of the SIMPLE method for predicting incompressible flows”, *Numerical Heat Transfer*, Vol. 7, 147–163, 1984.

[7] J. P. Van Doormaal and G. D. Rauthby, “An evaluation of the segregated approach for predicting incompressible fluid flows”, in *National Heat Transfer Conference*, Denver, Colorado, 1985.

[8] J. P. Van Doormaal, G. D. Raithby, and B. H. McDonald, “The segregated approach to predicting viscous compressible fluid flows”, *J. Turbomach*, Vol. 109, 268–277, 1987.

[9] Richard C. Martineau, “An efficient semi-implicit pressure based scheme employing a high resolution finite element method for simulating transient and steady, inviscid and viscous, compressible flows on unstructured grids”, *PhD Thesis*, University of Idaho, USA, 2002.

[10] A. J. Chorin, “Numerical methods for solving incompressible viscous flows”, *J. Comp. Phys.*, Vol. 2, 12–26, 1967.

[11] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator splitting”, *J. Comp. Phys.*, Vol. 62, 40–65, 1985.

[12] R. I. Issa, A. D. Gosman, and A. P . Watkins, “The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme”, *J. Comp. Phys.*, Vol. 62, 66–82, 1985.

[13] P. J. Oliveira and R. I. Issa, “An improved PISO algorithm for the computation of buoyancy-driven flows”, *Numer. Heat Transfer Part B*, Vol. 40, 473–493, 2001.

[14] C. M. Rhei and W. L. Chow, “A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation”, *AIAA Journal*, Vol. 21, 1525–1536, 1983.

[15] S. Abdallah, “Numerical solutions for the pressure Poisson equation with Neumann boundary conditions using a non-staggered grid, I”, *J. Comp. Phys.*, Vol. 70, 182–192, 1987.

[16] S. Abdallah, “Numerical solutions for the pressure Poisson equation with Neumann boundary conditions using a non-staggered grid, II”, *J. Comp. Phys.*, Vol. 70, 193–202, 1987.

[17] F. Sotiropoulos and S. Abdallah, “The discrete cintinuity equation in primitive variable solutions of incompressible flows”, *J. Comp. Phys.*, Vol. 95, No. 1, 212–227, 1991.

[18] P. A. Russel, “Numerical solutions for the incompressible Navier-Stokes equations in primitive variables with low Reynolds number flow applications”, *PhD Thesis*, University of Cincinnati, 1997.

[19] R. I. Issa, B. Ahamadi-Befrui, K. R. Beshay, and A. D. Gosman, “Solution of implicitly discretized reacting flow equations by operator-splitting”, *J. Comp. Phys.*, Vol. 93, 388–410, 1991.

[20] Habib N. Najm, Peter S. Wyckoff, and Omar M. Knio, “A semi implicit numerical scheme for reacting flow. I. Stiff chemistry”, *J. Comput. Phys*, Vol. 143, 381–402, 1998.

[21] Omar M. Knio, Habib N. Najm, and Peter S. Wyckoff, “A semi implicit numerical scheme for reacting flow. II. Stiff, operator-split formulation”, *J. Comput. Phys*, Vol. 154, 428–467, 1999.

[22] R. Hilbert, F. Tap, H. El-Rabii, and D. Thevenin, “Impact of detailed chemistry and transport models on turbulent combustion simulations”, *Prog. Energy Combust. Sci.*, Vol. 30, 61–117, 2004.

[23] L. Douglas Smoot, “A decade of combustion research”, *Prog. Energy Combust. Sci.*, Vol. 23, 203–232, 1997.

[24] B. Bedat, F. Egolfopoulos, and T. Poinsot, “Direct numerical simulation of heat release and NOx formation in turbulent non premixed flames”, *Combust. Flame*, Vol. 119, No. 1/2, 69–83, 1999.

[25] L. Vervisch and T. Poinsot, “Direct numerical simulation of non-premixed turbulent flames”, *Ann. Rev. Fluid Mech.*, Vol. 30, 655–691, 1998.

[26] Veynante D. and Vervisch L., “Turbulent combustion modelling”, *Prog. Energy Combust. Sci.*, Vol. 28, 193–266, 2002.

[27] S. Mahalingham, J. H. Chen, and L. Vervisch, “Finite rate chemistry and transient effects in direct numerical simulation of turbulent nonpremixed flames”, *Combust. Flame*, Vol. 102, No. 3, 285–297, 1995.

[28] Carolyn R. Kaplan, Seung W. Baek, Elaine S. Oran, and Janet L. Ellzey, “Dynamics of a strongly radiating unsteady ethylene jet diffusion flame”, *Combust. Flame*, Vol. 96, 1–21, 1994.

[29] P. J. Coehlo, O. J. Teerling, and D. Roekaerts, “Spectral radiative effects and turbulence/radiation interaction in a non-luminous turbulent jet diffusion flame”, *Combust. Flame*, Vol. 133, 75–91, 2003.

[30] K. Liu and R. H. Pletcher, “A procedure to establish inflow conditions for LES of spatially developing turbulent boundary layers”, in *Proceedings of HT-FED2004*, Charlote, North Carolina, USA, July 11-15, 2004.

[31] M. D. Smooke, C. S. Mcenally, L. D. Pfefferle, R. J. Hall, and M. B. Colket, “Computational and experimental study of soot formation in coflow, laminar diffusion flame”, *Combust. Flame*, Vol. 117, 117–139, 1999.

[32] C. S. McEnally and L. D. Pfefferle, “Aromatic and linear hydrocarbon concentration measurements in non-premixed flame”, *Combust. Sci. and Technol.*, Vol. 116-117, 183–209, 1996.

[33] O. Oymak, “Method of lines solution of time-dependent 2D Navier-Stokes

equations for incompressible separated flows", *PhD Thesis*, Middle East Technical University, Turkey, 1997.

[34] O. Oymak and N. Selçuk, "Method of Lines solution of time-dependent two-dimensional Navier-Stokes equations", *Int. J. Numer. Meth. Fluids*, Vol. 23, No. 5, 455–466, 1996.

[35] O. Oymak and N. Selçuk, "Transient simulation of internal separated flows using an intelligent higher-order spatial discretization scheme", *Int. J. Numer. Meth. Fluids*, Vol. 24, 759–769, 1997.

[36] N. Selçuk and O. Oymak, "A novel code for the prediction of transient flow field in a gas turbine combustor simulator", in *NATO/RTO Meeting Proceedings 14, AVT Symposium on Gas Turbine Engine Combustion, Emissions and Alternative Fuels*, 1–10, 1999.

[37] T. Tarhan, "Numerical simulation of flow through inlet flues of heat recovery steam generators", *Master's Thesis*, Middle East Technical University, Turkey, 1999.

[38] T. Tarhan and N. Selçuk, "Method of lines for transient flow fields", *Int. J. Comput. Fluid Dynamics*, Vol. 15, 309–328, 2001.

[39] C. Erşahin, "Parallelization of a transient Navier-Stokes code based on method of lines solution", *Master's Thesis*, Middle East Technical University, Turkey, 2001.

[40] N. Selçuk, T. Tarhan, and S. Tanrikulu, "Comparison of method of lines and finite difference solutions of 2D Navier-Stokes equations for transient laminar pipe flow", *Int. J. Numer. Meth. Engng.*, Vol. 53, No. 7, 1615–1628, 2002.

[41] A. B. Uygur, "Numerical simulation of transient turbulent flow in a heated pipe", *Master's Thesis*, Middle East Technical University, Turkey, 2002.

[42] C. Ersahin, T. Tarhan, I. H. Tuncer, and N. Selçuk, "Parallelization of a transient method of lines Navier Stokes code", *Int. J. Comput. Fluid Dynamics*, Vol. 18, No. 1, 81–92, 2004.

[43] A. Bilge Uygur, T. Tarhan, and N. Selçuk, "MOL solution for transient turbulent flow in a heated pipe", *Int. J. Therm. Sci.*, Vol. 44, No. 8, 726–734, 2005.

[44] Y. Xu and M. D. Smooke, "Application of primitive variable Newton's method

for the calculation of axisymmetric laminar diffusion flame", *J. Comp. Phys.*, Vol. 109, 99–109, 1993.

[45] R. E. Mitchell, "Nitrogen oxide formation in laminar methane-air diffusion flames", *PhD Thesis*, MIT, 1975.

[46] R. E. Mitchell, A. F. Sarofim, and L. A. Clomburg, "Experimental and numerical investigation of confined laminar diffusion flames", *Combust. Flame*, Vol. 37, 227–244, 1980.

[47] A. M. Shehata and D. M. McEligot, "Mean structure in the viscous layer of strongly-heated internal gas flows. Measurements", *Int. J. Heat Mass Trans.*, Vol. 41, 4297–4313, 1998.

[48] S. Satake, T. Kunugi, A. M. Shehata, and D. M. McEligot, "Direct numerical simulation for laminarization of turbulent forced gas flows in circular tubes with strong heating", *Int. J. Heat Fluid Flow*, Vol. 21, 526–534, 2000.

[49] S. Harmandar and N. Selçuk, "The method of line solution of discrete ordinated method for radiative heat transfer in cylindrical enclosures", *JQSRT*, Vol. 84, No. 4, 409–422, 2004.

[50] F. N. Çayan, "The method of lines solution of discrete ordinates method for non-gray media", *Master's Thesis*, Middle East Technical University, 2006.

[51] Y. Morinishi, O. V. Vasilyev, and T. Ogi, "Fully conservative finite difference scheme in cylindrical coordinates for incompressible flow simulations", *J. Comp. Phys.*, Vol. 197, 686–710, 2004.

[52] E.E. Khalil, J. B. Spalding, and J. H. Whitelaw, "The calculation of local flow properties in two-dimensional furnaces", *Int. J. Heat Mass Transfer*, Vol. 18, 775–791, 1975.

[53] C. K. Westbrook and F. L. Dryer, "Simplified reaction mechanisms for the oxidation of hydrocarbon fuels in flames", *Combust. Sci. and Technol*, Vol. 27, 31–43, 1981.

[54] J. Hsu and S. Mahalingam, "Performances of reduced reaction mechanisms in unsteady non-premixed flame simulations", *Combust. Theory Modelling*, Vol. 7, 365–382, 2003.

[55] R.J. Kee, F.M. Rupley, E. Meeks, and J.A. Miller, "CHEMKIN-III: A

FORTRAN Chemical Kinetics Package for the Analysis of Gas Phase Chemical and Plasma Kinetics", Tech. Rep. SAND96-8216, Sandia Laboratories, 1996.

[56] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, and J.A. Miller, "A FORTRAN Computer Code Package for the Evaluation of Gas-Phase, Multicomponent Transport Properties,", Tech. Rep. SAND86-8646, Sandia Laboratories, 1986.

[57] R. J. Kee, F. M. Rupley, and J. A. Miller, "The CHEMKIN thermodynamics database", Tech. Rep. SAND87-8215B, Sandia National Laboratories, 1996.

[58] R. Bird, W. Stewart, and E. Lightfoot, *Transport Phenomena*. John Wiley and Sons, Inc.,, 2 ed., 2002.

[59] W. E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*. New York: Academic Press, 1991.

[60] F. N. Çayan and N. Selçuk, "The method of lines solution of discrete ordinates method for non-grey media", *J. Quant. Spectrosc. Radiat. Transfer*, Vol. 104, No. 2, 228–237, 2007.

[61] J. D. Anderson, *Computational Fluid Dynamics. The Basics with Applications*. Singapore: McGraw-Hill, 1995.

[62] N. Selçuk, A . Bilge Uygur, I. Ayranci, and T. Tarhan, "Transient simulation of radiating flows", *J. Quant. Spectrosc. Radiat. Transfer*, Vol. 93, No. 3, 151–161, 2005.

[63] T. Tarhan and N. Selçuk, "Investigation of difference schemes on Method of Lines", *Progress in Computational Fluid Dynamics*, Vol. 6, No. 8, 447–458, 2007.

[64] I. Ayranci and N. Selçuk, "MOL solution of DOM for transient radiative transfer in 3–D scattering media", *J. Quant. Spect. Rad. Trans.*, Vol. 84, 409–422, 2004.

[65] R. J. LeVeque, *Numerical Methods for Conservation Laws*. Basal: Birkhauser Verlag, 1992.

[66] C. Hirsch, *Numerical Computation of Internal and External Flows*, Vol. 2. Chichester: John Wiley & Sons, 1998.

[67] R. Weiner, B. A. Schmitt, and H. Podhaisky, “ROWMAP – a ROW–code with Krylov techniques for large stiff ODEs”, tech. rep., FB Mathematik und Informatik, Universitaet Halle, 1996.

[68] K. Radhakrishnan and A. C. Hindmarsh, “Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations”, tech. rep., Lawrance Livermore National Laboratory, NASA, 1993.

[69] P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, 2001.

[70] J. C. Adams, “MUDPACK: Multigrid portable FORTRAN software for the efficient solution of linear elliptic partial differential equations”, *Appl. Math. Comput.*, Vol. 34, 113–146, 1989.

[71] “http://www.cisl.ucar.edu/css/software/mudpack/ (last accessed on 7/2/2007).”

[72] B. Lessani and M. V. Papalexandris, “Time-accurate calculation of variable density flows with strong temperature gradients and combustion”, *J. Comp. Phys.*, Vol. 212, 218–246, 2006.

[73] B. G. Carlson and K. D. Lathrop, “Transport theory - The method of discrete ordinates”, in *Computing methods in reactor physics* (H. Greenspan, C.N. Kelber, and D. Okrent, Eds.), 171–266, New York: Gordon and Breach, 1968.

[74] J. Hyde and J. S. Truelove, “The discrete ordinates approximation for multidimensional radiant heat transfer in furnaces”, Tech. Rep. HTFS-RS-189, AERE-R8502, AERE Harwell, 1977.

[75] A. Yücel, “Solution of the discrete ordinates equations for a radiatively participating medium by the method of lines”, *Advances in Computer Methods for Partial Differential Equations VII*, 838–844, 1992.

[76] B. Leckner, “Spectral and total emissivity of water vapor and carbon dioxide”, *Combust. Flame*, Vol. 19, 33–48, 1978.

[77] M. F. Modest, “The weighted-sum-of-gray-gases model for arbitrary solution methods in radiative transfer”, *ASME Journal of Heat Transfer*, Vol. 113, 650–656, 1991.

[78] H. C. Hottel and A. F. Sarofim, *Radiative transfer*. New York: McGraw-Hill, 1967.

[79] M. K. Denison and W. A. Fiveland, “A correlation for the reordered wave number of the wide-band absorptance of radiating gases”, *ASME Journal of Heat Transfer*, Vol. 119, 853–856, 1997.

[80] M. K. Denison and B. W. Webb, “A spectral line-based weighted-sum-of-gray gases model for arbitrary RTE solvers”, *ASME Journal of Heat Transfer*, Vol. 115, 1004–1012, 1993.

[81] AMTEC Engineering, Inc., *TECPLOT, version 10.0*, 2003.

[82] A. M. Shehata and D. M. McEligot, “Turbulence structure in the viscous layer of strongly heated gas fows”, Tech. Rep. INEL-95/0223, Idaho National Engineering Laboratory, 1995.

[83] K. Ezato, A. M. Shehata, T. Kunugi, and D. M. McEligot, “Numerical prediction of transitional features of turbulent forced gas flows in circular tubes with strong heating”, *ASME Journal of Heat Transfer*, Vol. 141, 546–555, 1999.

[84] D. P. Mikielewicz, A. M. Shehata, J. D. Jackson, and D. M. McEligot, “Temperature, velocity and mean turbulence structure in strongly heated internal gas flows: Comparison of numerical predictions with data”, *Int. J. Heat Mass Transfer*, Vol. 45, 4333–4352, 2002.

[85] T. Tarhan and N. Selçuk, “A novel CFD code based on method of lines for reacting flows : Verification on methane/air diffusion flame”, *Combust. Sci. and Tech.*, Vol. 179, 39–60, 2007.

[86] Ziji Zhang, “Theoretical and computational study of coupling of soot, gas kinetics and radiation in diffusion flames using reduced mechanisms”, *PhD Thesis*, The University of Texas at Austin, 1998.

APPENDIX A

ORDINATES AND WEIGHTS FOR S_N APPROXIMATIONS

Table A.1: Discrete ordinates for the S_N approximation for axisymmetric cylindrical geometry.

Order of approximation	Ordinates			Weights
	μ_m	η_m	ξ_m	w_m
S_2	0.5000000	0.7071068	0.5000000	3.1415927
S_4	0.2958759	0.2958759	0.9082483	1.0471976
	0.2958759	0.9082483	0.2958759	1.0471976
	0.9082483	0.2958759	0.2958759	1.0471976
S_6	0.1838670	0.1838670	0.9656013	0.3219034
	0.1838670	0.6950514	0.6950514	0.7252938
	0.6950514	0.1838670	0.6950514	0.7252938
	0.1838670	0.9656013	0.1838670	0.3219034
	0.6950514	0.6950514	0.1838670	0.7252938
	0.9656013	0.1838670	0.1838670	0.3219034
S_8	0.1422555	0.1422555	0.9795543	0.3424718
	0.1422555	0.5773503	0.8040087	0.1984568
	0.5773503	0.1422555	0.8040087	0.1984568
	0.1422555	0.8040087	0.5773503	0.1984568
	0.5773503	0.5773503	0.5773503	0.9234358
	0.8040087	0.1422555	0.5773503	0.1984568
	0.1422555	0.9795543	0.1422555	0.3424718
	0.5773503	0.8040087	0.1422555	0.1984568
	0.8040087	0.5773503	0.1422555	0.1984568
	0.9795543	0.1422555	0.1422555	0.3424718

APPENDIX B

INPUT FILES

B.1 CHEM.INP

```
ELEMENTS C H O N END

SPECIES CH4 O2 H2O CO2 N2 END

REACTIONS

END
```

B.2 MESHGEN.INI

```
================================================================================
================================== 1 ===========================================
================================================================================
NR    NZ     Rs          Rf          Zs         Zf          aR       aZ
================================================================================
41   101    0.0000D0   0.5000d0    0.0000D0  1.5000D1    0.50D0  0.50D0
********************************************************************************
********************************************************************************
********************************************************************************

================================================================================
================================== 2 ===========================================
================================================================================
NR    NZ     Rs          Rf          Zs         Zf          aR       aZ
================================================================================
65   129    0.0000D0   1.3700d0    0.0000D0  1.000D2    0.50D0  0.50D0
********************************************************************************
********************************************************************************
********************************************************************************

================================================================================
================================== 3 ===========================================
================================================================================
NR    NZ     Rs          Rf          Zs         Zf          aR       aZ
================================================================================
65   129    0.0000D0   2.5400d0    0.0000D0  3.000D1    0.50D0  0.50D0
********************************************************************************
********************************************************************************
********************************************************************************
```

B.3 LOCATIONS.INI

```
================================================================================
                        Output locations - Test Case 1
================================================================================
 radial tolerance      :         0.10    !  [rtol]      ! cm
 radial location 1     :         0.00    !  [r_1]       ! cm
 radial location 2     :         0.20    !  [r_2]       ! cm
 radial location 3     :         0.30    !  [r_3]       ! cm
 radial location 4     :         0.40    !  [r_4]       ! cm
 radial location 5     :         0.45    !  [r_5]       ! cm
 axial tolerance       :         0.40    !  [ztol]      ! cm
 axial  location 1     :        28.77    !  [l_1]       ! cm
 axial  location 2     :         0.00    !  [l_2]       ! cm
 axial  location 3     :        58.91    !  [l_3]       ! cm
 axial  location 4     :         0.00    !  [l_4]       ! cm
 axial  location 5     :        87.13    !  [l_5]       ! cm
 ================================================================================

 ================================================================================
                        Output locations - Test Case 2
================================================================================
 radial tolerance      :         0.1     !  [rtol]      ! cm
 radial location 1     :         0.0     !  [r_1]       ! cm
 radial location 2     :         0.2     !  [r_2]       ! cm
 radial location 3     :         0.3     !  [r_3]       ! cm
 radial location 4     :         0.4     !  [r_4]       ! cm
 radial location 5     :         0.5     !  [r_5]       ! cm
 axial tolerance       :         0.4     !  [ztol]      ! cm
 axial  location 1     :         1.2     !  [l_1]       ! cm
 axial  location 2     :         0.0     !  [l_2]       ! cm
 axial  location 3     :         2.4     !  [l_3]       ! cm
 axial  location 4     :         0.0     !  [l_4]       ! cm
 axial  location 5     :         5.0     !  [l_5]       ! cm
 ================================================================================
```

B.4 DATA.INI

```
=============================================================================================================
 Definitions of the parameters               :       value    !  parameter      ! unit/explanation
=============================================================================================================
 program ID                                  :           3    !  [progid]       ! Methane-Air Diffusion Flame
=============================================================================================================
 Mesh related
=============================================================================================================
 orientation                                 :           1    !  [igeom]        ! 0=horizontal, 1=vertical
 number of subdomains                        :           1    !  [ndomain]      ! -
 grid set to be used from meshgen.ini        :           3    !  [gridset]      ! see meshgen.ini
 total number of grid points in r-direction  :          65    !  [m]            ! irp*(2**(ier-1))+1
 total number of grid points in z-direction  :         129    !  [n]            ! jzq*(2**(jez-1))+1
 fuel inlet grid location                    :           1    !  [ibottom]      ! -
 fuel inlet grid location                    :          17    !  [itop]         ! -
=============================================================================================================
 Algorithm related
=============================================================================================================
 formulation                                 :           2    !  [ipressure]    ! see bottom of the page
 pressure equation solved by                 :           2    !  [ipsoln]       ! 1=FISHPACK, 2=MUDPACK
 number of momentum phases                   :           2    !  [nstage_m]     ! -
 number of energy phases                     :           2    !  [nstage_e]     ! -
 number of species phases                    :           2    !  [nstage_s]     ! -
=============================================================================================================
 Time integration related
=============================================================================================================
 initial data                                :           1    !  [incond]       ! 0=rest, 1=input file
 initial time                                :      0.5d+0    !  [t0]           ! s
 final time                                  :      0.6d+0    !  [tf]           ! s
 time step                                   :      1.0d-5    !  [delt]         ! s
 window time                                 :      5.0d-2    !  [twind]        ! s
 number of frames                            :         100    !  [nframe]       ! -
 frequency of desired instant results        :           5    !  [ip_freq]      ! @ ip_freq*delt intervals
=============================================================================================================
 ODE solver related
=============================================================================================================
 odesolver                                   :           2    !  [iodesolv]     ! 1=ROWMAP, 2=LSODES, 3=E-F, 4=E-C
 length of array  work used in odesolver     :    10000000    !  [lrw]          ! -
 length of array iwork used in odesolver     :          90    !  [liw]          ! -
 relative tolerance rtol for odesolver       :      1.0d-4    !  [rtol]         ! -
 absolute tolerance atol for odesolver       :      1.0d-4    !  [atol]         ! -
=============================================================================================================
 Poisson solver related
=============================================================================================================
 max. number of multi-grid cycles            :           1    !  [maxcy]        ! -
 method of relaxation                        :           0    !  [method]       ! 0=point, 3=line relax. in r&z
 order of approximation                      :           4    !  [morder]       ! -
=============================================================================================================
 Flow related
=============================================================================================================
 Reynolds number                             :      36.3d0    !  [reynolds]     ! based on fuel inlet velocity
 reference temperature                       :    298.0d+0    !  [t_ref]        ! K
 reference pressure                          :  1.01325d+6    !  [p_ref]        ! dyn/cm2
=============================================================================================================
 Reaction related
=============================================================================================================
 fuel inlet velocity                         :     4.50d+0    !  [u_fuel]       ! cm/s
 oxidizer inlet velocity                     :     9.88d+0    !  [u_oxid]       ! cm/s
 heat of reaction                            :   5.0016d+4    !  [Q_heat]       ! ergs/g
 average specific heat                       :   1.5980d+0    !  [cp_heat]      ! ergs/g.K
 reaction mechanism                          :           1    !  [mech_no]      ! see bottom of the page
 chemkin linking files path                  : object/lstep/  !  [chemkin_path] ! -
=============================================================================================================
 Radiation related
=============================================================================================================
 radiation                                   :           1    !  [irad]         ! 0=none, 1=exists
 radiation to flow grid ratio in r           :           8    !  [rfr]          !
 radiation to flow grid ratio in z           :           8    !  [rfz]          !
 order of approximation                      :           4    !  [ns_rad]       !
 initial time for radiation                  :      0.0d0     !  [t0_rad]       !
 final time for radiation                    :     10.0d0     !  [tf_rad]       !
 print time for radiation                    :       1.d0     !  [tp_rad]       !
 error for ode solver                        :      1.0d-3    !  [errode_rad]   !
 error for convergence                       :      2.0d0     !  [errconv_rad]  !
 initial condition for radiation             :      1.0d2     !  [xi0_rad]      ! W/m2-sr
 component indicator                         :          3     !  [n_comp]       ! 1=water,2=CO2,3=mixtr
 total pressure indicator                    :          1     !  [n_pres]       ! -
 total pressure                              :      1.0d0     !  [p_total]      ! bar
 mol fraction of water vapor                 :      1.0d0     !  [xh2o]         ! -
 mol fraction of carbon dioxide              :      0.0d0     !  [xco2]         ! -
 emissivity @ z=0 (wall 2)                   :      1.0d0     !  [ebd_rad2]     ! -
 emissivity @ r=R (wall 3)                   :      1.0d0     !  [ebd_rad3]     ! -
 emissivity @ z=L (wall 4)                   :      1.0d0     !  [ebd_rad4]     ! -
=============================================================================================================

Explanations:
-------------------------------------------------------------------------------------------------------------
[mech_no]    |   reaction mechanisms                                                 !  chemkin linking files path
-------------------------------------------------------------------------------------------------------------
   [-1]      |   non-reacting flow                                                   ! 1-step/
   [ 0]      |   flame-sheet                                                         ! 1-step/
   [ 1]      |   1-step rxn. mech. by Khalil et al. (1975).                          ! 1-step/
   [ 2]      |   1-step rxn. mech. by Westbrook and Dryer, (1981)                    ! 1-step/
   [ 3]      |   1-step rxn. mech. by Bui-Pham, (1992) (Thesis of Hsu)               ! 1-step/
   [ 4]      |   2-step rxn. mech. by Pember et al., (1998)                          ! 2-step/
   [ 5]      |   5-step rxn. mech. by Mallampalli et al., (1996)  (Chen)             ! 5-step-chen/
   [ 6]      |  10-step rxn. mech. by Chen, (1997)                                   !10-step-chen/
   [ 7]      |   full chemistry                                                      !gri-mech-1.2/
-------------------------------------------------------------------------------------------------------------
[ipressure]  | Formulation of the governing equations
-------------------------------------------------------------------------------------------------------------
   [ 1]      |  Non-conservative & incompressible
   [ 2]      |  Non-conservative & compressible
   [ 3]      |  Conservative & compressible
-------------------------------------------------------------------------------------------------------------
```

APPENDIX C

SOURCE CODES

C.1 Program COMBINE

```
!##########################################################################
!##                                                                      ##
!##   This program is for post processing of output files                ##
!##   Originally  developed by Tanil Tarhan,   October 2003              ##
!##   Modified by A. Bilge Uygur,              June 2005                 ##
!##                                                                      ##
!##########################################################################

!##########################################################################
      module common_header
      implicit none
!##########################################################################
!...
!...  double precision
      integer, parameter :: prec  = 8  !(double)
!...
!...  number of subdomains
      integer :: ndomain
      integer, allocatable, dimension(:) :: ndom
!...
!...  number of grid points in r- and z-direction in main domain
      integer :: m
      integer :: n
!...
!...  integer parameters
      integer :: i, j, k, ifile
      integer :: is, nv, izone, nvar
      integer :: noin, nocb, nost, nosy, noex, nws, nwe
      integer :: status
      integer :: ich4,io2,ih2o,ico2,ico,in2
      integer :: icode,igeom,mech_no,progid
      integer :: ier,irp,jzq,jez
!...
!...  name of the files
      character (len =   2) :: char_nos
      character (len = 120) :: inputfile
      character (len =  16) :: name_unused
!...
!...  variables to be read from the files
      real (kind=prec), allocatable, dimension(:,:,:) :: variables
      character(len =  16), allocatable, dimension(:) :: name_read
!...
!##########################################################################
      end module common_header
!##########################################################################
```

```
!########################################################################
      program output_analysis
      use common_header
      implicit none
!########################################################################
!...
!...  prompt for the type of execution carried out
      write (*,1000)
      write (*,1001) '[1] :  code : sequential | results: all     '
      write (*,1001) '[2] :  code : parallel   | results: all     '
      write (*,1001) '[3] :  code : sequential | results: instant '
      write (*,1001) '[4] :  code : parallel   | results: instant '
      write (*,1002)
      read  (*,1003) icode
!...
!...  read input data from file [../data.in]
      call read_parameters
      if (igeom == 0) &
      write(*,*)'outputs will be formed for horizontal geometry'
      if (igeom == 1) &
      write(*,*)'outputs will be formed for vertical geometry'
      if (progid < 3) &
      write(*,*) 'species will be excluded from data files'
      write(*,1002)
!...
!...  open input and output files
      call open_files
!...
!...  if instant results are to be produced then set izone to 1
      if( icode == 3) then
      izone = 1
      goto 100
      endif
      if( icode == 4) then
      izone = 1
      goto 100
      endif
!...
!...  if transient results are to be produced enter the number of zone
      write(*,*) '(?) enter the number of zone ='
      read (*,*) izone
!...
!...  read all the zone
100   continue
      if( icode == 1) call seq_combine
      if( icode == 2) call par_combine
      if( icode == 3) call seq_combine
      if( icode == 4) call par_combine
!...
!...  extract final zone from combined data
      call extract_steady_state
!...
!...  extract major variable
      if (progid == 3) then
      call extract_major_variables
      endif
!...
!...  create the symmetric output file
      call create_symmetry
!...
!...  formats
1000  format (74('-')/,1x,'SELECT ONE OF THE FOLLOWINGS:'&
             ,/,74('-'))
1001  format (1x,A)
1002  format (74('-'))
1003  format(i10)
!########################################################################
      end program output_analysis
```

```
!########################################################################



!########################################################################
      subroutine read_parameters
      use common_header
      implicit none
!########################################################################
      character (len=100) :: datafile
      datafile = '../data.ini'
!...
!...  open data.in to read necessary information
      open (51,file=datafile)
!...
!========================================================================
!...  read from data.in
!========================================================================
      do i=1,3
      read (51,*)         ! # of lines to skip to reach progid
      enddo
      read (51,100) progid
      write(*,*)progid
      do i=1,3
      read (51,*)         ! # of lines to skip to reach igeom
      enddo
      read (51,100) igeom
      write(*,*)igeom
      read (51,100) ndomain
      write(*,*)ndomain
      read (51,*)         ! skip 1 line to reach grids
      read (51,100) m
      write(*,*)m
      read (51,100) n
      close (51)
!...
!========================================================================
!...  formats
100   format(46x,1i16)
!...
!########################################################################
      return
      end subroutine
!########################################################################



!########################################################################
      subroutine open_files
      use common_header
      implicit none
!########################################################################
      goto(100,200,300,400) icode
!========================================================================
!     sequential code output files (dom0all.plt)
!========================================================================
100   continue
      i = 0
      write (char_nos, '(I1)') i
      inputfile = 'dom' // trim(char_nos) // 'all.plt'
!...
!...  set the number of the files
      noin = 10
      nocb = 11
      nost = 12
      nosy = 13
      noex = 14
```

```
      open (noin,file=inputfile)
      open (nocb,file='seq_comb.plt')
      open (nost,file='seq_stst.plt')
      open (nosy,file='seq_symm.plt')
      if (progid == 3) open (noex,file='mjr_stst.plt')
      goto 500
!...
!=======================================================================
!     parallel code output files (domXall.plt)
!=======================================================================
200   continue
      do i=1,ndomain
!...
!...  convert the nos number to character to use in nameing the files
      if (i <= 9) write (char_nos, '(I1)') i
      if (i >  9) write (char_nos, '(I2)') i
!...
!...  set the names of the files
      inputfile = 'dom' // trim(char_nos) // 'all.plt'
!...
!...  set the number of the files
      noin  = 15 + i
!...
!...  open the input files
      open (noin,file=inputfile)
      enddo
!...
!...  open the output files
      nocb = 11
      nost = 12
      nosy = 13
      noex = 14
      open (nocb,file='par_comb.plt')
      open (nost,file='par_stst.plt')
      open (nosy,file='par_symm.plt')
      if (progid == 3) open (noex,file='mjr_stst.plt')
      goto 500
!...
!=======================================================================
!     instant sequential code output files (output0.plt)
!=======================================================================
300   continue
      i = 0
      write (char_nos, '(I1)') i
      inputfile  = 'output' // trim(char_nos) // '.plt'
!...
!...  open the output files
      noin = 10
      nocb = 11
      nost = 12
      nosy = 13
      noex = 14
      open (noin,file=inputfile)
      open (nocb,file='seq_comb.plt')
      open (nost,file='seq_stst.plt')
      open (nosy,file='seq_symm.plt')
      if (progid == 3) open (noex,file='mjr_stst.plt')
      goto 500
!...
!=======================================================================
!     instant parallel code output files (outputX.plt)
!=======================================================================
400   continue
      do i=1,ndomain
!...
!...  convert the nos number to character to use in nameing the files
      if (i <= 9) write (char_nos, '(I1)') i
      if (i >  9) write (char_nos, '(I2)') i
!...
```

```
!...  set the names of the files
      inputfile  = 'output' // trim(char_nos) // '.plt'
!...
!...  set the number of the files
      noin  = 15 + i
!...
!...  open the input files
      open (noin,file=inputfile)
      enddo
!...
!...  open the output files
      nocb = 11
      nost = 12
      nosy = 13
      noex = 14
      open (nocb,file='par_comb.plt')
      open (nost,file='par_stst.plt')
      open (nosy,file='par_symm.plt')
      if (progid == 3) open (noex,file='mjr_stst.plt')
      goto 500
!=======================================================================
!...  continue
500   continue
!#######################################################################
      return
      end subroutine open_files
!#######################################################################



!#######################################################################
      subroutine seq_combine
      use common_header
      implicit none
!#######################################################################
      write (*,1000)
!...
!...  read the number of variables from the input file
      read (noin,300) nvar
!...
!...  allocate read arrays
      allocate (variables(nvar,m,n), stat=status)
      allocate (name_read(nvar)    , stat=status)
!...
!...  read the name of the variables
      read (noin,*) name_unused,name_unused,(name_read(k), k=1,nvar)
!...
!=======================================================================
!...  non-reacting flow initiation
!=======================================================================
      if (progid < 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nocb,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" '
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nocb,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" '
      endif
      endif
!...
!=======================================================================
!...  reacting flow initiation
!=======================================================================
      if (progid == 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nocb,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" "s"'&
```

```
                  ,(' "',name_read(k),'" ',k=9,nvar)
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nocb,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" "s"'&
                  ,(' "',name_read(k),'" ',k=9,nvar)
      endif
      endif
!...
!=======================================================================
!...  read all zone information from input file
!=======================================================================
      do k=1,izone
      write (*,1003) k
!...
!...  skip the line containing "zone" in the input file
      read (noin,400)
!...
!...  write the zone information to the output file
      write (nocb,100) n,m
!...
!...  read data from input file and write to ouput file
      do i=1,m
      do j=1,n
      read  (noin,200)  (variables(nv,i,j), nv=1,nvar)
      enddo
!...
!...  write data to output file
      do j=1,n
      write (nocb,200)  (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
      enddo
      close(noin)
!...
!=======================================================================
!...  output file created: print its name
!=======================================================================
      write (*,1002) ' seq_comb.plt'
      write (*,1001)
!...
!=======================================================================
!...  formats
90    format (A,100A)
100   format (10h zone   i=,i4,2h, ,6h    j=,i4,2h, ,10h f=point  )
200   format (100e16.5)
300   format (16x,i4)
400   format (10x,i4)
1000  format (74('-')/,1x,'Combining process started'&
              ,/,74('-'))
1001  format (74('-')/,1x,'Combining process ended'&
              ,/,74('-'))
1002  format (1x,'---> output =',A)
1003  format (1x,'---> zone   =',1i4)
!#######################################################################
      return
      end subroutine seq_combine
!#######################################################################




!#######################################################################
      subroutine par_combine
      use common_header
      implicit none
!#######################################################################
      write (*,1000)
!...
```

```
!...  read the number of variables from the input file
      do i=1,ndomain
      noin = 15 + i
      read (noin,300) nvar
      enddo
!...
!...  allocate all the arrays
      allocate (     ndom(ndomain ), stat=status)
      allocate (variables(nvar,m,n), stat=status)
      allocate (name_read(nvar)    , stat=status)
!...
!...  read the name of the variables
      do i=1,ndomain
      noin = 15 + i
      read (noin,*) name_unused,name_unused,(name_read(k), k=1,nvar)
      enddo
!...
!=====================================================================
!...  non-reacting flow initiation
!=====================================================================
      if (progid < 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nocb,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" '
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nocb,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" '
      endif
      endif
!...
!=====================================================================
!...  reacting flow initiation
!=====================================================================
      if (progid == 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nocb,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" "s"'&
                     ,(' "',name_read(k),'" ',k=9,nvar)
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nocb,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" "s"'&
                     ,(' "',name_read(k),'" ',k=9,nvar)
      endif
      endif
!...
!=====================================================================
!...  read all zone information from input file
!=====================================================================
      zone : do k=1,izone
      write (*,1003) k
!...
!...  write the zone information to the output file
      write (nocb,100) n,m
!...
!...  read the number of grid point of the current domain
      do i=1,ndomain
      noin = 15 + i
      read (noin,400) ndom(i)
      enddo
!...
!...  read data from each input file and write to output file
      do i=1,m
      do is=1,ndomain
      noin = 15 + is
      nws  = 4
      nwe  = ndom(is)-3
      if(is == 1      ) nws = 1
```

```
      if(is == ndomain) nwe = ndom(is)
!...  read data from input file
      do j=1,ndom(is)
      read  (noin,200)  (variables(nv,i,j), nv=1,nvar)
      enddo
!...  write data to output file
      do j=nws,nwe
      write (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
      enddo
      enddo zone
      close(noin)
!...
!=======================================================================
!...  output file created: print its name
!=======================================================================
      write (*,1002) ' par_comb.plt'
      write (*,1001)
!...
!=======================================================================
!...  formats
90    format (A,100A)
100   format (10h zone   i=,i4,2h, ,6h    j=,i4,2h, ,10h f=point  )
200   format (100e16.5)
300   format (16x,i4)
400   format (10x,1i4)
1000  format (74('-')/,1x,'Combining process started'&
              ,/,74('-'))
1001  format (74('-')/,1x,'Combining process ended'&
              ,/,74('-'))
1002  format (1x,'---> output =',A)
1003  format (1x,'---> zone   =',1i4)
!#######################################################################
      return
      end subroutine par_combine
!#######################################################################



!#######################################################################
      subroutine extract_steady_state
      use common_header
      implicit none
!#######################################################################
      write (*,1000)
!...
!=======================================================================
!...  read data from combined file
!=======================================================================
      rewind (nocb)
      read (nocb,*)
!...
      do k=1,izone-1
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,*)
      enddo
      enddo
      write (*,1003) k
      enddo
!...
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
```

```
      enddo
      write (*,1004) izone
!...
!=======================================================================
!...  non-reacting flow initiation
!=======================================================================
      if (progid < 3) then
!...
      write (nost,80) '# of variables:',nvar
!...
      if (igeom == 0) then     !horizontal orientation
      write (nost,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" '
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nost,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" '
      endif
      endif
!...
!=======================================================================
!...  reacting flow initiation
!=======================================================================
      if (progid == 3) then
!...
      write (nost,80) '# of variables:',nvar
!...
      if (igeom == 0) then     !horizontal orientation
      write (nost,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" "s"'&
                    ,(' "',name_read(k),'" ',k=9,nvar)
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nost,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" "s"'&
                    ,(' "',name_read(k),'" ',k=9,nvar)
      endif
      endif
!...
!=======================================================================
!...  write down the variables in tecplot format
!=======================================================================
      write (nost,100) n,m
      do i=1,m
      do j=1,n
      write (nost,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
!...
!=======================================================================
!...  output file created: print its name
!=======================================================================
      if( icode == 1 ) write (*,1002) ' seq_stst.plt'
      if( icode == 2 ) write (*,1002) ' par_stst.plt'
      write (*,1001)
!...
!=======================================================================
!...  formats
80    format (A,i4)
90    format (A,100A)
100   format (10h zone   i=,i4,2h, ,6h    j=,i4,2h, ,10h f=point  )
200   format (100e16.5)
1000  format (74('-')/,1x,'Steady state results are being extracted'&
              ,/,74('-'))
1001  format (74('-')/,1x,'Steady state results were formed'        &
              ,/,74('-'))
1002  format (1x,'---> output =',A)
1003  format (1x,'---> zone   =',1i4,2x,'skipped')
1004  format (1x,'---> zone   =',1i4,2x,'is being extracted')
!#######################################################################
      return
```

```
      end subroutine extract_steady_state
!#############################################################################




!#############################################################################
      subroutine extract_major_variables
      use common_header
      implicit none
!#############################################################################
!...
!...  this subroutine extracts the variables of interest if there
!...  are many and is used for reacting flows
      rewind(nost)
!...
!...  start combining
      write (*,1000)
!...
!...  read first line from each input file
      read (nost,10) nvar
      write(*,*) nvar
!...
!...  read the name of the variables
      read (nost,*) name_unused,name_unused,(name_read(k), k=1,nvar)
!...
!...  identify the species indices
      call major_species_identifier
      write(*,*)'after major_species_identifier'
!...
!=============================================================================
!...  reacting flow initiation
!=============================================================================
      if (igeom == 0) then
      write (noex,*) ' variables = "z" "r" "u" "v" "ch4" "o2"  &
                     "h2o" "co2" "n2" "t" "p" "divq" "s" '
      endif
      if (igeom == 1) then
       write (noex,*) ' variables = "r" "z" "u" "v" "ch4" "o2" &
                     "h2o" "co2" "n2" "t" "p" "divq" "s" '
      endif
      write (noex,100) n,m
!...
!=============================================================================
!...  read from the input file and write to output file
!=============================================================================
!...  read a line from each input file
      read (nost,400)
!...
!...  read from domXall.dat and write to
      do i=1,m
      do j=1,n
      read  (nost,200)  (variables(nv,i,j), nv=1,nvar)
      enddo
!...  write to fort.91
      do j=1,n
      write (noex,200) (variables(nv,i,j), nv=1,4) &
                      ,variables(ich4,i,j),variables( io2,i,j) &
                      ,variables(ih2o,i,j),variables(ico2,i,j) &
                      ,variables( in2,i,j),variables(   5,i,j) &
                      ,variables(   6,i,j),variables(   7,i,j) &
                      ,variables(   8,i,j)
      enddo
      enddo
      close (nost)
      close (noex)
!...
!=============================================================================
!...  output file created: print its name
```

```
!========================================================================
      write (*,1002) ' steadyst.plt'
      write (*,1001)
!...
!========================================================================
!...  formats
10    format (15x,i4)
80    format (A,i4)
90    format (A,100A)
100   format (10h zone   i=,i3,2h, ,6h    j=,i3,2h, ,10h f=point  )
110   format (10h zone   i=,i3,2h, ,10h f=point  )
200   format (100e16.5)
400   format (10x,1i3)
1000  format (74('-')/,1x,'Extraction process started',/,74('-'))
1001  format (74('-')/,1x,'Extraction process ended',/,74('-'))
1002  format (1x,'---> output =',A)
!...
!########################################################################
      end subroutine extract_major_variables
!########################################################################



!########################################################################
      subroutine create_symmetry
      use common_header
      implicit none
!########################################################################
!...
!...  mirror image creation starts
      write (*,1000)
!...
!========================================================================
!...  non-reacting flow initiation
!========================================================================
      if (progid < 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nosy,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" '
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nosy,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" '
      endif
      endif
!...
!========================================================================
!... reacting flow initiation
!========================================================================
      if (progid == 3) then
!...
      if (igeom == 0) then     !horizontal orientation
      write (nosy,90) 'variables = "z" "r" "u" "v" "t" "p" "divq" "s"'&
                     ,('"',name_read(k),'" ',k=9,nvar)
      endif
!...
      if (igeom == 1) then     !vertical orientation
      write (nosy,90) 'variables = "r" "z" "u" "v" "t" "p" "divq" "s"'&
                     ,('"',name_read(k),'" ',k=9,nvar)
      endif
      endif
!...
      rewind (nocb)
      read  (nocb,*)
!...
!========================================================================
!========================================================================
!...                 Mirror image for non-reacting flows
```

```
!=======================================================================
!=======================================================================
!..
      if (progid < 3) then
!...
!=======================================================================
!...  for horizontal configuration
!=======================================================================
      if (igeom == 0) then
      zone_symhnr : do k=1,izone
      write (*,1003) k
!...
!...  read from combined.plt
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
!...
!...  write to symmetry.plt
      write (nosy,100) n,m*2
      do i=m,1,-1
      do j=1,n
      write (nosy,200)  variables(1,i,j),-variables(2,i,j), &
                        variables(3,i,j), variables(4,i,j), &
                        variables(5,i,j), variables(6,i,j), &
                        variables(7,i,j)
      enddo
      enddo
      do i=1,m
      do j=1,n
      write (nosy,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
      enddo zone_symhnr
      close (nosy)
      close (nocb)
      endif
!...
!=======================================================================
!...  for vertical configuration
!=======================================================================
      if (igeom == 1) then
      zone_symvnr : do k=1,izone
      write (*,1003) k
!...
!...  read from combined.plt
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
!...
!...  write to symmetry.plt
      write (nosy,100) n,m*2
      do i=m,1,-1
      do j=1,n
      write (nosy,200) -variables(1,i,j), variables(2,i,j), &
                        variables(3,i,j), variables(4,i,j), &
                        variables(5,i,j), variables(6,i,j), &
                        variables(7,i,j)
      enddo
      enddo
      do i=1,m
      do j=1,n
      write (nosy,200) (variables(nv,i,j), nv=1,nvar)
      enddo
```

```
      enddo
      enddo zone_symvnr
      close (nosy)
      close (nocb)
      endif
      endif
!...
!=======================================================================
!=======================================================================
!...                 Mirror image for reacting flows
!=======================================================================
!=======================================================================
!...
      if (progid == 3) then
!...
!=======================================================================
!...  for horizontal configuration
!=======================================================================
      if (igeom == 0) then
      zone_symhr : do k=1,izone
      write (*,1003) k
!...
!...  read from combined.plt
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
!...
!...  write to symmetry.plt
      write (nosy,100) n,m*2
      do i=m,1,-1
      do j=1,n
      write (nosy,200)  variables(1,i,j),-variables(2,i,j), &
                        variables(3,i,j), variables(4,i,j), &
                       (variables(nv,i,j), nv=5,nvar)
      enddo
      enddo
      do i=1,m
      do j=1,n
      write (nosy,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
      enddo zone_symhr
      close (nosy)
      close (nocb)
      endif
!...
!=======================================================================
!...  for vertical configuration
!=======================================================================
      if (igeom == 1) then
      zone_symvr : do k=1,izone
      write (*,1003) k
!...
!...  read from combined.plt
      read (nocb,*)
      do i=1,m
      do j=1,n
      read (nocb,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
!...
!...  write to symmetry.plt
      write (nosy,100) n,m*2
      do i=m,1,-1
      do j=1,n
      write (nosy,200) -variables(1,i,j), variables(2,i,j), &
```

```
                      variables(3,i,j), variables(4,i,j), &
                     (variables(nv,i,j), nv=5,nvar)
      enddo
      enddo
      do i=1,m
      do j=1,n
      write (nosy,200) (variables(nv,i,j), nv=1,nvar)
      enddo
      enddo
      enddo zone_symvr
      close (nosy)
      close (nocb)
      endif
      endif
!...
!=====================================================================
!...  mirror creation ends: print output file names
!=====================================================================
      if( icode == 1 ) write (*,1002) ' seq_symm.plt'
      if( icode == 2 ) write (*,1002) ' par_symm.plt'
      write (*,1001)
!...
!=====================================================================
!...  formats
80    format (A,i4)
90    format (A,100A)
100   format(10h zone   i=,i3,2h, ,6h    j=,i3,2h, ,10h f=point  )
200   format(100e16.5)
1000  format (74('-')/,1x,'Symmetric output creation started'&
              ,/,74('-'))
1001  format (74('-')/,1x,'Symmetric output were formed'&
              ,/,74('-'))
1002  format (1x,'---> output =',A)
1003  format (1x,'---> zone   =',1i4)
!#####################################################################
      return
      end subroutine create_symmetry
!#####################################################################



!#####################################################################
      subroutine major_species_identifier
!...  This subroutine determines the indices for major species
      use common_header
      implicit none
!#####################################################################
      do k=1,nvar
      if(name_read(k) == 'CH4') ich4 = k
      if(name_read(k) ==  'O2')  io2 = k
      if(name_read(k) == 'H2O') ih2o = k
      if(name_read(k) == 'CO2') ico2 = k
      if(name_read(k) ==  'CO')  ico = k
      if(name_read(k) ==  'N2')  in2 = k
      enddo
!...
!#####################################################################
      return
      end subroutine major_species_identifier
!#####################################################################
```

C.2 Program EXTRACT

```
!########################################################################
!##                                                                    ##
!##   This program is for post processing of output files.             ##
!##   It extracts axial and radial profiles from the output file       ##
!##                                                                    ##
!##   Originally  developed by Tanil Tarhan,   October 2003            ##
!##   Modified by A. Bilge Uygur,              June 2005               ##
!##                                                                    ##
!########################################################################

!########################################################################
      program extract
      implicit none
!########################################################################
!...
!...  double precision
      integer, parameter :: prec  = 8  !(double)
!...
!...  integer parameters
      character (len=100) :: inputfile,outputfile,locations,name_unused
      integer :: nr, nz ,i, j, k, ivar,igeom,itype,mech_no,progid
      integer :: nofin,nofout,nofl,status,ierror
      integer :: nz_1, nz_2, nz_3, nz_4, nz_5,          &
                 nr_1, nr_2, nr_3, nr_4, nr_5
      real (kind=prec) :: tolr,tolz
      real (kind=prec) :: z_1, z_2, z_3, z_4, z_5,      &
                          r_1, r_2, r_3, r_4, r_5
      real (kind=prec) :: zf_1, zf_2, zf_3, zf_4, zf_5, &
                          rf_1, rf_2, rf_3, rf_4, rf_5
!...
!...  variables to be read from the files
      character (len=100) :: datafile
      character (len=16), allocatable, dimension(:)     :: name_read
      real (kind=prec),   allocatable, dimension(:,:,:) :: variables
!...
!########################################################################
!...
!...  datafiles
      datafile   = '../data.ini'
      locations  = 'locations.ini'
      outputfile = 'profiles.plt'
      nofin  = 11
      nofout = 12
      nofl   = 13
      open (nofout,file=outputfile)
      open (51, file=datafile)
      open (nofl, file=locations)

!...  read the necessary information from data.in
!...  read the physical orientation of the system
      do i=1,3   !# of lines to skip in data.in for progid
      read (51,*)
      enddo
      read (51,1000) progid
      write(*,*)progid
      do i=1,3   !# of lines to skip in data.in for igeom
      read (51,*)
      enddo
      read (51,1000) igeom
      write (*,*) igeom
!...  decide on the type of execution by reading ndomain from data.in
      read (51,1000) itype
      write (*,*) itype
!...
!...  read the axial and radial locations which data is to be
```

```
!...  extracted at
!...
      read (nofl,*)
      read (nofl,*)
      read (nofl,*)
      read (nofl,999)tolr
      read (nofl,999)r_1
      read (nofl,999)r_2
      read (nofl,999)r_3
      read (nofl,999)r_4
      read (nofl,999)r_5
      read (nofl,999)tolz
      read (nofl,999)z_1
      read (nofl,999)z_2
      read (nofl,999)z_3
      read (nofl,999)z_4
      read (nofl,999)z_5
      close(nofl)
!...
!...  set the inputfile
      if (itype == 1) inputfile  = 'seq_stst.plt'
      if (itype == 2) inputfile  = 'par_stst.plt'
      open (nofin ,file=inputfile )
!...
!...  read the number of variables in the input file
      read (nofin,1001) ivar
!...
!...  first skip the line containing the variable names
      read (nofin,*)
!...
!...  read the number of grid point in z and r directions
      read (nofin,1002) nz,nr
!...
!...  alllocate the array in which all variables will be stored
      allocate (variables(ivar,nr,nz), stat=status)
      allocate (name_read(ivar)      , stat=status)
!...
!...  go to the head of the file
      rewind nofin
!...
!...  read the variable names
      read (nofin,*)
      read (nofin,*) name_unused,name_unused,(name_read(k), k=1,ivar)
      read (nofin,*)
!...
!...  read all variables
      do i=1,nr
      do j=1,nz
      read (nofin,1003)  (variables(k,i,j), k=1,ivar)
      enddo
      enddo
!...
!=======================================================================
!...                          locate data locations
!=======================================================================
!...  initialize the locations
      nr_1 = nr+1
      nr_2 = nr+1
      nr_3 = nr+1
      nr_4 = nr+1
      nr_5 = nr+1
      nz_1 = nz+1
      nz_2 = nz+1
      nz_3 = nz+1
      nz_4 = nz+1
      nz_5 = nz+1
!...
!=======================================================================
!...  radial locations
```

```
!=======================================================================
!...
!...  if the physical orientation is horizontal
      if (igeom == 0) then
      do i=1,nr
      if ((variables(2,i,1) <= r_1+tolr) .and.  &
      (variables(2,i,1) >=  r_1-tolr)) then
      nr_1 = i ; rf_1 = variables(2,nr_1,1)
      endif
      if ((variables(2,i,1) <= r_2+tolr) .and.  &
      (variables(2,i,1) >=  r_2-tolr)) then
      nr_2 = i ; rf_2 = variables(2,nr_2,1)
      endif
      if ((variables(2,i,1) <= r_3+tolr) .and.  &
      (variables(2,i,1) >=  r_3-tolr)) then
      nr_3 = i ; rf_3 = variables(2,nr_3,1)
      endif
      if ((variables(2,i,1) <= r_4+tolr) .and.  &
      (variables(2,i,1) >=  r_4-tolr)) then
      nr_4 = i ; rf_4 = variables(2,nr_4,1)
      endif
      if ((variables(2,i,1) <= r_5+tolr) .and.  &
      (variables(2,i,1) >=  r_5-tolr)) then
      nr_5 = i ; rf_5 = variables(2,nr_5,1)
      endif
      enddo
      endif
!...
!...  if the physical orientation is vertical
      if (igeom == 1) then
      do i=1,nr
      if ((variables(1,i,1) <= r_1+tolr) .and.  &
      (variables(1,i,1) >=  r_1-tolr)) then
      nr_1 = i ; rf_1 = variables(1,nr_1,1)
      endif
      if ((variables(1,i,1) <= r_2+tolr) .and.  &
      (variables(1,i,1) >=  r_2-tolr)) then
      nr_2 = i ; rf_2 = variables(1,nr_2,1)
      endif
      if ((variables(1,i,1) <= r_3+tolr) .and.  &
      (variables(1,i,1) >=  r_3-tolr)) then
      nr_3 = i ; rf_3 = variables(1,nr_3,1)
      endif
      if ((variables(1,i,1) <= r_4+tolr) .and.  &
      (variables(1,i,1) >=  r_4-tolr)) then
      nr_4 = i ; rf_4 = variables(1,nr_4,1)
      endif
      if ((variables(1,i,1) <= r_5+tolr) .and.  &
      (variables(1,i,1) >=  r_5-tolr)) then
      nr_5 = i ; rf_5 = variables(1,nr_5,1)
      endif
      enddo
      endif
!...
!=======================================================================
!...  locate axial data locations
!=======================================================================
!...
!...  if the physical orientation is horizontal
      if (igeom == 0) then
      do j=1,nz
      if ((variables(1,1,j) <= z_1+tolz) .and.  &
      (variables(1,1,j) >=  z_1-tolz)) then
      nz_1 = j ; zf_1 = variables(1,1,nz_1)
      endif
      if ((variables(1,1,j) <= z_2+tolz) .and.  &
      (variables(1,1,j) >=  z_2-tolz)) then
      nz_2 = j ; zf_2 = variables(1,1,nz_2)
      endif
```

```
      if ((variables(1,1,j) <= z_3+tolz) .and.  &
      (variables(1,1,j) >=  z_3-tolz)) then
      nz_3 = j ; zf_3 = variables(1,1,nz_3)
      endif
      if ((variables(1,1,j) <= z_4+tolz) .and.  &
      (variables(1,1,j) >=  z_4-tolz)) then
      nz_4 = j ; zf_4 = variables(1,1,nz_4)
      endif
      if ((variables(1,1,j) <= z_5+tolz) .and.  &
      (variables(1,1,j) >=  z_5-tolz)) then
      nz_5 = j ; zf_5 = variables(1,1,nz_5)
      endif
      enddo
      endif
!...
!...  if the physical orientation is vertical
      if (igeom == 1) then
      do j=1,nz
      if ((variables(2,1,j) <= z_1+tolz) .and.  &
      (variables(2,1,j) >=  z_1-tolz)) then
      nz_1 = j ; zf_1 = variables(2,1,nz_1)
      endif
      if ((variables(2,1,j) <= z_2+tolz) .and.  &
      (variables(2,1,j) >=  z_2-tolz)) then
      nz_2 = j ; zf_2 = variables(2,1,nz_2)
      endif
      if ((variables(2,1,j) <= z_3+tolz) .and.  &
      (variables(2,1,j) >=  z_3-tolz)) then
      nz_3 = j ; zf_3 = variables(2,1,nz_3)
      endif
      if ((variables(2,1,j) <= z_4+tolz) .and.  &
      (variables(2,1,j) >=  z_4-tolz)) then
      nz_4 = j ; zf_4 = variables(2,1,nz_4)
      endif
      if ((variables(2,1,j) <= z_5+tolz) .and.  &
      (variables(2,1,j) >=  z_5-tolz)) then
      nz_5 = j ; zf_5 = variables(2,1,nz_5)
      endif
      enddo
      endif
!...
!=====================================================================
!...  check the validity of the locations
!=====================================================================
75    if (nr_1 == nr+1) then
      goto 77
      elseif (nr_2 == nr+1) then
      goto 77
      elseif (nr_3 == nr+1) then
      goto 77
      elseif (nr_4 == nr+1) then
      goto 77
      elseif (nr_5 == nr+1) then
      goto 77
      endif
!...
76    if (nz_1 == nz+1) then
      goto 78
      elseif (nz_2 == nz+1) then
      goto 78
      elseif (nz_3 == nz+1) then
      goto 78
      elseif (nz_4 == nz+1) then
      goto 78
      elseif (nz_5 == nz+1) then
      goto 78
      endif
!...
!...  if the locations are successfully located proceed
```

```
      goto 79
!...
77    write(*,*)'Your tolerance in r direction does not conform  &
                with your mesh'
      write(*,*)'Check domXmesh.inp and revise your tolerance'
      write(*,*)'Output file could not been created !!!'
      write(*,1016)
      goto 1050
!...
78    write(*,*)'Your tolerance in z direction does not conform  &
                with your mesh'
      write(*,*)'Check domXmesh.inp and revise your tolerance'
      write(*,*)'Output file could not been created !!!'
      write(*,1016)
      goto 1050
!...
79    continue
!...  echo the locations and corresponding grids
      write(*,1016)
      write(*,1015)'Radial locations to be found and their  &
                    corresponding grids are:'
      write(*,1016)
      write(*,1017) r_1,nr_1,rf_1
      write(*,1017) r_2,nr_2,rf_2
      write(*,1017) r_3,nr_3,rf_3
      write(*,1017) r_4,nr_4,rf_4
      write(*,1017) r_5,nr_5,rf_5
      write(*,1016)
      write(*,1015)'Radial locations to be found and their  &
                    corresponding grids are:'
      write(*,1016)
      write(*,1017) z_1,nz_1,zf_1
      write(*,1017) z_2,nz_2,zf_2
      write(*,1017) z_3,nz_3,zf_3
      write(*,1017) z_4,nz_4,zf_4
      write(*,1017) z_5,nz_5,zf_5
      write(*,1016)
!...
!=========================================================================
!...  write down the profiles
!=========================================================================
!...
!...  inititation for non-reacting flows
      if (progid <3) then
      if (igeom == 0) then
      write (nofout,1004) 'variables = "z" "r" "u" "v" "t" "p" "divq" '
      endif
      if (igeom == 1) then
      write (nofout,1004) 'variables = "r" "z" "u" "v" "t" "p" "divq" '
      endif
      endif
!...
!...  inititation for reacting flows
      if (progid == 3) then
      if (igeom == 0) then
      write (nofout,1004) 'variables = "z" "r" "u" "v" "t" "p"     &
                    "divq" "s" ',('"',name_read(k),'" ',k=9,ivar)
      endif
      if (igeom == 1) then
      write (nofout,1004) 'variables = "r" "z" "u" "v" "t" "p"     &
                    "divq" "s" ',('"',name_read(k),'" ',k=9,ivar)
      endif
      endif
!...
!...  write down the profiles to the file
!...  profiles @ radial station 1
      write (nofout,1005) nz
      i=nr_1
      do j=1,nz
```

```
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ radial station 2
      write (nofout,1006) nz
      i=nr_2
      do j=1,nz
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ radial station 3
      write (nofout,1007) nz
      i=nr_3
      do j=1,nz
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ radial station 4
      write (nofout,1008) nz
      i=nr_4
      do j=1,nz
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ radial station 5
      write (nofout,1009) nz
      i=nr_5
      do j=1,nz
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ axial station 1
      write (nofout,1010) nr
      j=nz_1
      do i=1,nr
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ axial station 2
      write (nofout,1011) nr
      j=nz_2
      do i=1,nr
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ axial station 3
      write (nofout,1012) nr
      j=nz_3
      do i=1,nr
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ axial station 4
      write (nofout,1013) nr
      j=nz_4
      do i=1,nr
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  profiles @ axial station 5
      write (nofout,1014) nr
      j=nz_5
      do i=1,nr
      write (nofout,1003)  (variables(k,i,j), k=1,ivar)
      enddo
!...
!...  echo the user creation of the output file
      write(*,*)'Output file profiles.plt is successfully created'
      write(*,1016)
```

```
!...
!========================================================================
!...  formats
999   format (46x,1f16.5)
1000  format (46x,1i16)
1001  format (15x,i5)
1002  format (10x,i4,8x,i4)
1003  format (100e16.5)
1004  format (A,100A)
1005  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = r_1')
1006  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = r_2')
1007  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = r_3')
1008  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = r_4')
1009  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = r_5')
1010  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = z_1')
1011  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = z_2')
1012  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = z_3')
1013  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = z_4')
1014  format ('zone',2x,'i=',i4,4x,'f=point',2x,'t = z_5')
1015  format (1x,A)
1016  format (1x,74('-'))
1017  format (1f7.2,' cm corresponding to',i4,' th grid  and ',  &
              1f7.2,' cm on the mesh' )
1018  format (5(3x,i4))
!########################################################################
1050  stop
      end
!########################################################################
```

APPENDIX D

REFERENCE TRANSIENT SOLUTIONS

The transient axial velocity and temperature fields obtained with the previously developed code by Uygur *et al.* [2] are displayed in Figures D.1-D.2 as a reference unsteady solution for the laminar methane-air diffusion flame problem under consideration.

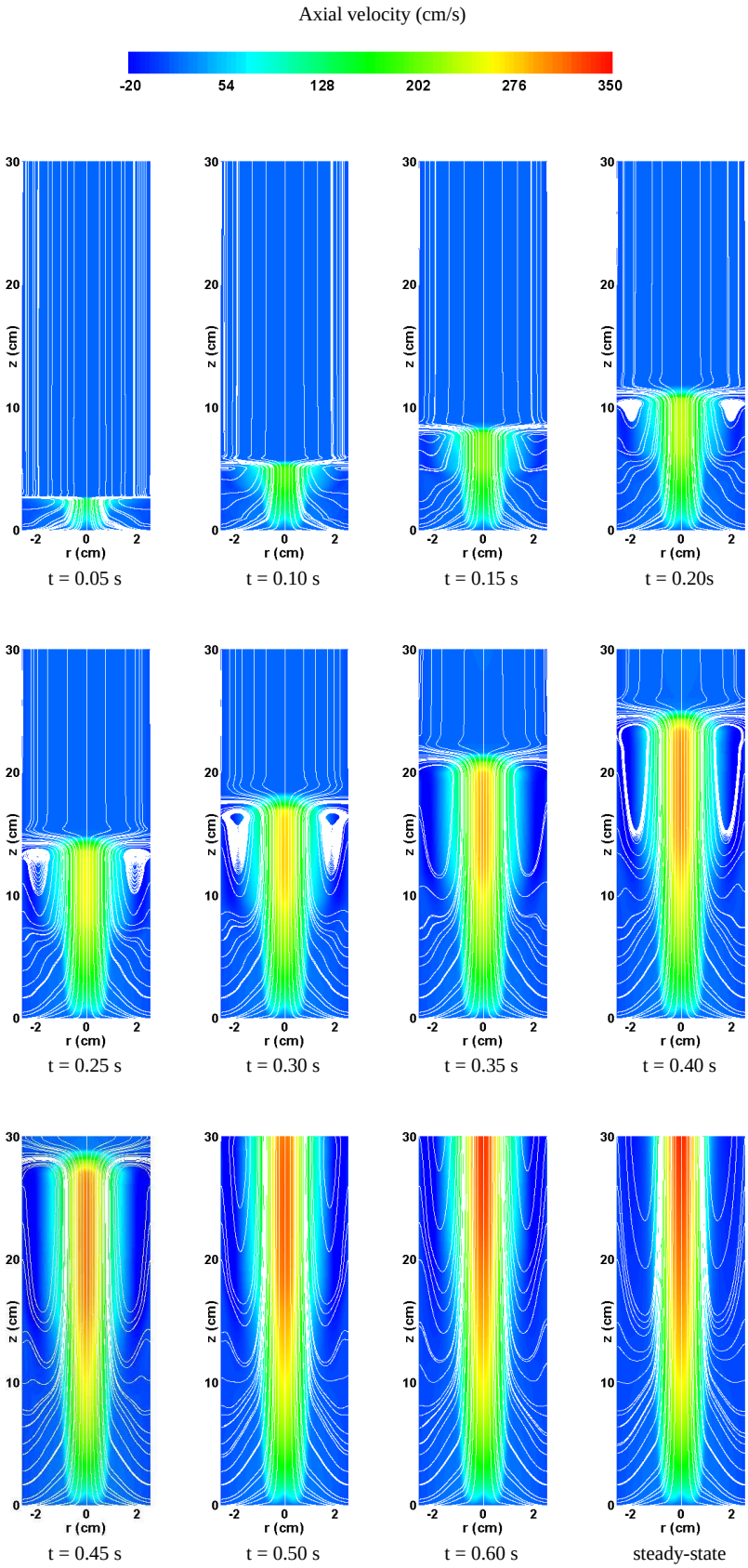


Figure D.1: Time development of streamline pattern and axial velocity obtained by Uygur *et al.* [2].

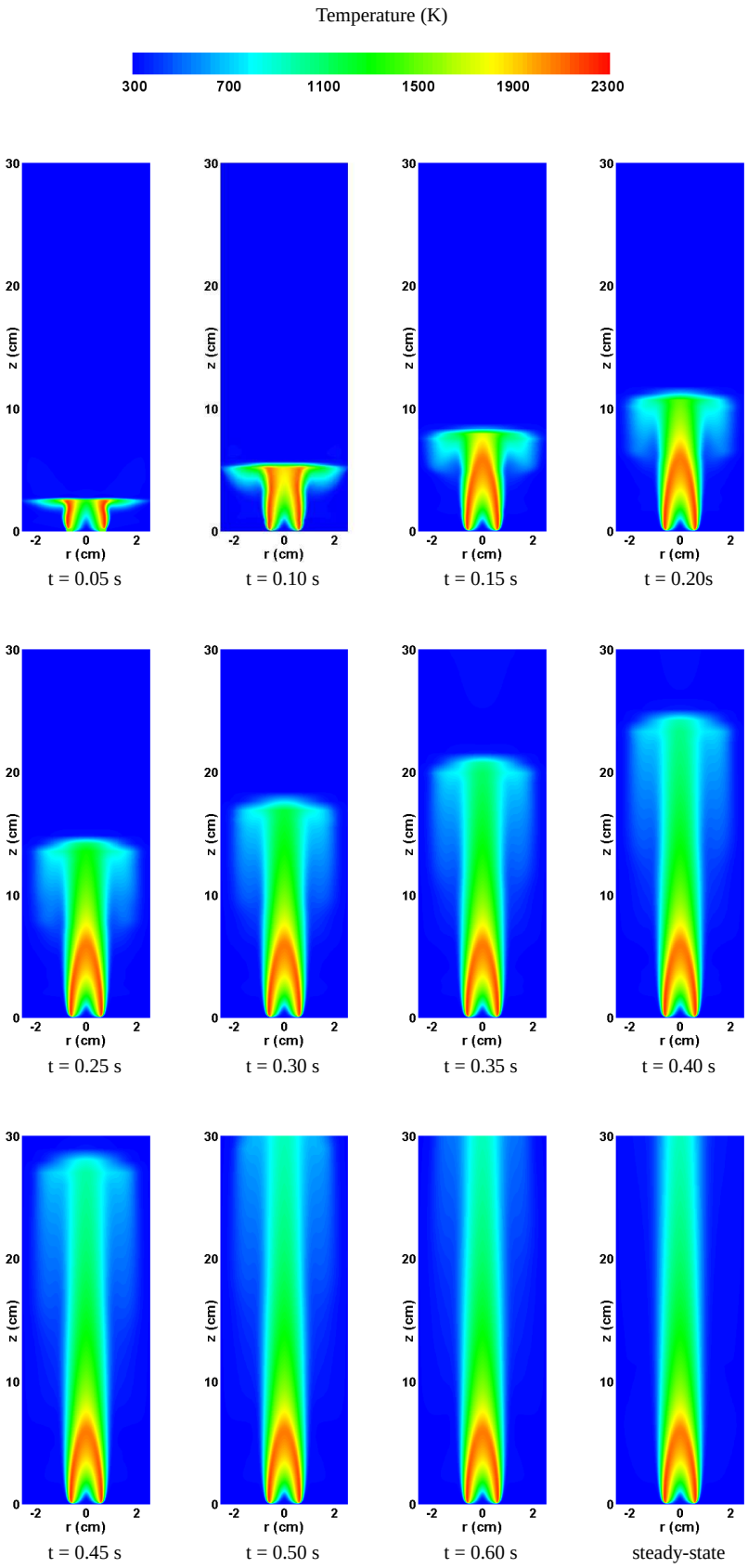


Figure D.2: Time development of temperature field obtained by Uygur *et al.* [2].

CURRICULUM VITAE

EDUCATION

- **Ph.D.** (Chemical Engineering)
  [September 2002 – March 2007]
  Middle East Technical University, Department of Chemical Engineering, Ankara, Turkey
  Dissertation Title: A Non-iterative Pressure Based Algorithm for the Computation of Reacting Radiating Flows

- **M.Sc.** (Chemical Engineering)
  [September 2000 – August 2002]
  Middle East Technical University, Department of Chemical Engineering, Ankara, Turkey
  Dissertation Title: Numerical Simulation of Transient Turbulent Flow in a Heated Pipe

- **B.Sc.** (Chemical Engineering)
  [September 1996 – June 2000]
  Middle East Technical University, Department of Chemical Engineering, Ankara, Turkey

PROFESSIONAL EXPERIENCE:

- **Teaching Assistant**
  [December 2000 – December 2004]
  Middle East Technical University, Department of Chemical Engineering, Ankara, Turkey

- **Department computer administrator**
  [December 2004 – To date]
  Middle East Technical University, Department of Chemical Engineering, Ankara, Turkey

ACADEMIC EXPERIENCE:

- Attendee to THIMUN (The Hague International Model United Nations) as a delegate from Turkey representing Tarsus American College, Den Hague, The Netherlands 1994.
- Attendee to "2nd Chemical Engineering Conference for Collaborative Research in Eastern Mediterranean", Ankara, Turkey, May 2001.
- Attendee to "The Third International Symposium on Radiation Transfer, RAD-01", Antalya, Turkey, June 2001.
- Attendee to the panel on "Steam-Electricity Generation from Fluidized Bed Combustion" organized by TÜBİTAK, MİMAG-SAMKO and CMEC, Ankara, Turkey, December 2001.
- Attendee to "International Symposium on Turbulence Heat And Mass Transfer, THMT-04", Antalya, Turkey, October 2003.
- Attendee to "The Fourth International Symposium on Radiation Transfer, RAD-04", İstanbul, Turkey, June 2004.
- Assistantship to undergraduate and graduate courses namely, Heat and Mass Transfer Operations, Fluid Mechanics, Chemical Engineering Laboratory, Thermodynamics, Mathematical Modeling, Chemical Engineering Design, Combustion Technology.

AREAS OF EXPERTISE

- Computational fluid dynamics, Direct Numerical Simulation, Combustion, Numerical analysis, Parallel scientific computation, Beowulf cluster set-up and administration, Linux operating systems.

COMPUTER RELATED EXPERIENCE AND SKILLS

- **Experience:** Department Computer Coordinator, High performance scientific computation on Linux Beowulf clusters, System administration of Beowulf clusters, Apache WEB server and Bind DNS server administration.
- **Programming Languages:** FORTRAN 77/90, HTML, PHP
- **Operating Systems:** Windows platforms, Unix, Advanced level Linux (RedHat,SUSE, Mandriva, Debian)
- **Commercial and Research Software Packages:** Tecplot, Sigmaplot, CHEMKIN, CHEMCAD, PVM3
- **Others:** LaTeX, BibTeX, Postscript, Microsoft Office

PUBLICATIONS

Journal Articles

- N. Selçuk, A.B. Uygur, I. Ayranci and T. Tarhan, “Transient simulation of radiating flows", J. Quant. Spectrosc. Radiat. Transfer, Vol. 93(1-3), pp. 151-161, 2005.
- A.B. Uygur, T. Tarhan and N. Selçuk, “MOL solution for transient turbulent flow in a heated pipe", Int. J. Therm. Sci., Vol. 44, pp. 726-734, 2005.
- A. Bilge Uygur, Tanil Tarhan and Nevin Selçuk, “Transient simulation of reacting radiating flows, Int. J. Therm. Sci., Vol. 65, pp. 969-976, 2006.
- A. Bilge Uygur, Nevin Selçuk and I. Hakki Tuncer, “A non-iterative pressure based scheme for the computation of reacting radiating flows", Int. J. Therm. Sci. (in press)

Conference and Symposium Proceedings

- A.B. Uygur, T. Tarhan and N. Selçuk, "MOL solution for transient turbulent flow in a heated pipe", Proceedings of International Symposium on Advances in Computational Heat Transfer, Norway, April 19-24, 2004.
- N. Selçuk, A.B. Uygur, I. Ayranci and T. Tarhan, “Transient simulation of radiating flows", Proceedings of the Fourth International Symosium on Radiative Transfer, Istanbul, Turkey, June 20-25, 2004.

RESEARCH PROJECTS

- N. Selçuk, İ. H. Tuncer, T. Tarhan and A. B. Uygur, “Application of PARMOLS4MEE Computational Fluid Dynamics Code to Non-Reacting/Reacting Thermal Flow Problems", BAP No. 2002-03-04-08, METU, Ankara, June 2003.
- N. Selçuk, Y. Göğebakan, H. Altındağ, T. Tarhan and A. B. Uygur, “Determination of Combustion and Sulfur Capture Performance for Imported Coal and Imported Coal/Lignite Mixture in Fluidized Bed Combustor", AGUDOS No. 2003-03-04-04, METU, Ankara, October 2003.
- N. Selçuk, A. Bilge Uygur, “Numerical Simulation of Turbulent Reacting Flows", BAP No. 2004-07-02-00-125, METU, Ankara, July 2005.