The Numerical Solution of Reacting Flow Problems

by

Idrees Ahmad

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computer Studies
December 1998

Abstract

This thesis is concerned with the issue of finding an accurate, efficient and robust numerical solution technique for solving mathematical models of reactive flow. Two main issues of concern when solving these problems are large computational costs and numerical instabilities and inaccuracies. Over the past decades, many numerical techniques have been suggested for the solution of reacting flow problems. The work in this thesis is part of the continuing trend to find schemes which can solve reacting flow problems efficiently and robustly.

The technique described in this thesis uses the method of lines in which the underlying system of PDEs is discretized in space to give a system of ordinary differential equations in time. For time integration we have implemented the extended stability region method (NDF2) of Klopfenstein [47] and the analysis has shown that the integration step size has increased by a factor of 1.26. The spatial discretization is achieved by using flux limited finite difference and appropriate Riemann solver schemes in one and two space dimensions.

The use of the method of lines while solving PDEs governing both atmospheric dispersion as well as combustion problems results in a large system of ODEs which are highly coupled and stiff. In solving large stiff ODE system in time the approach of Verwer using a Gauss Seidel method for the stiff chemistry terms is extended to combustion problems. The method has the additional advantage that the storage requirement has been reduced considerably compared to conventional linear algebra methods.

The general trend while solving these ODEs is to control the local error per step. In this thesis we have extended the novel technique of controlling the local error per unit step with a time tolerance that varies with the spatial error. Techniques are developed for estimating the growth of the spatial error locally in time on model problems and then applied to both atmospheric and combustion problems. Numerical results are used to show that the approach works well, is automatic and can compare with the standard approach based on the CFL condition.

Acknowledgements

I would like to thank my supervisor Dr. Martin Berzins for his guidance and encouragement throughout the course of this research. Not only was he very helpful in guiding me throughout my stay at Leeds, but he was also very patient when it came to correcting my poorly drafted chapters as regards to the English language. Thanks also to Dr. Peter Jimack, Dr. David Hodgson, Dr. Victoria Pennington and Dr. A. Tomlin, for their helpful advices and discussions during this time.

My thanks also go to the General Office and Support staff of the School who were always happy to help me.

Thanks are also due to my colleagues Nasir Touheed, Muhammad Rafiq Asim, Zahid Hussain, Sharifullah Khan, Rashid Mahmood, Sarfraz Ahmad Nadeem, Allah Nawaz, Alam Zeb, Malik Muhammad Yousuf, Muhammad Shahid Saleem, Ahmad Saeed, Quadrat Ullaha, Muhammad Abbas and other colleague for matters not necessarily related to the research.

Members of my extended family in Pakistan and of my immediate family here in the UK have been a great encouragement throughout the period of this research. My thanks also to families of Mr. M. Ashraf and Mr. A. M. Chaudhary for related to family matters.

I especially wish to thank my parents (Mr. and Mrs. Hassan), brother, sisters, brother-in-laws, mother-in-law and father-in-law for their encouragement. This project would not have been completed without the constant support of my wife, Farrah Idrees.

Finally, my thanks also go to the Government of Pakistan for supporting me financially throughout my stay in Leeds.

At the very end I would like to thank The Almighty, for the much needed courage and strength which He granted me to finalise the project.

Nomenclature

a	Advection velocity
$\hat{A}$	Amplitude in trial solution used for the stability analysis
$\underline{A}$	Two dimensional mesh area
$\tilde{A}$, $\tilde{B}$	Parameter used to specify eigenvalues and eigenvectors and other related quantities along x-direction and y-direction
c	Courant number
$\tilde{c}$	Sound velocity
$c_{p,i}$	Specific heat at constant pressure of ith species
$c_{p,i}^{298}$	Specific heat at constant pressure of ith species at 298K
c_p	Specific heat at constant pressure
c_v	Specific heat at constant volume
$\hat{D}$	Diagonal matrix
$\mathbf{est}$	Spatial discretization error
$\hat{\mathbf{est}}$	Local growth in time of the spatial discretization error
e	Internal energy per unit mass
$e\hat{V}$	Error when approximation has been used to treat advection term explicitly
E	Total energy per unit volume
E_i	Emission sources for ith species
$\tilde{\mathbf{E}}^n$	Growth of temporal error at time t_n
$\hat{\mathbf{E}}^n$	PDE global error
E_a	Experimental activation energy
$\mathbf{F}(t_n, \mathbf{V}(t_n))$	Primary ODE function resulting from discretization of advection term and integration of the source term
$\tilde{\mathbf{F}}^f(t_n, \mathbf{V}(t_n))$	Discretization of advection term with second order upwind scheme
$\tilde{\mathbf{F}}^s(t_n, \mathbf{V}(t_n))$	Integration of source term with trapezoidal rule
$\mathbf{f}(\mathbf{u})$	Flux vector along x-direction
$\mathbf{G}(t_n, \mathbf{V}(t_n))$	Auxiliary ODE function to find auxiliary solution
$\mathbf{G}^f(t_n, \mathbf{V}(t_n))$	Discretization of advection term to find auxiliary solution
$\mathbf{G}^s(t_n, \mathbf{V}(t_n))$	Integration of source term to find auxiliary solution
$\mathbf{g}(\mathbf{u})$	Flux vector along y-direction

$\tilde{g}_{i_1}$	Dry deposition velocity of ith species
$\hat{g}_i$	Dimensionless factor represents the scavenging ratio of ith species
h_i	Specific enthalpy of ith species
h_i^f	Heat of formation of ith species
h	Specific enthalpy of a mixture
H_i	Total Enthalpy of ith species
H	Total enthalpy of a mixture
H_{mix}	Mixing height
$\tilde{j}$	Photolysis rate constant
I	Identity matrix
$\hat{I}$	Precipitation intensity
J	Jacobian matrix
J_s	Jacobian matrix of the source term
J_f	Jacobian matrix of the flow term
k_{i_1}	Dry deposition coefficient for ith species
k_{i_2}	Wet deposition coefficient for ith species
K_x	Diffusivity coefficient in x-direction
K_y	Diffusivity coefficient in y-direction
$\mathbf{le}(t_n)$	Time local error at time t_n
L	1D computational domain
$\hat{L}$	Lower triangular matrix
$L(w)$	Diagonal matrix of loss terms
$L^{(i)}$	ith left eigenvectors
$\tilde{m}$	Total moles of a mixture
$\tilde{m}_i$	Moles of ith species
$\tilde{M}$	Total mass of a mixture
$\tilde{M}_i$	Mass of ith species
M^*	The number of physical or chemical modelled considered in an atmospheric dispersion problem
NS	The number of species
p	Pressure
$\hat{p}_{V_c}$	Derivative of Roe-averaged pressure with respect

	to conserved variables
$\tilde{P}$	Wave number used in the trial solution for having stability analysis
P^*	The physical or chemical process being modelled in an atmospheric dispersion problem
re	Relative error in wave speed
P_i	Production term of ith species
q	Order of a method
R	Specific gas constant for a mixture of gases
$\tilde{R}$	Specific gas constant for the perfect gas
R_i	Specific gas constant for ith species
R_u	Universal gas constant
$R^{(i)}$	ith right eigenvectors
Re	Real part
S	Speed of the discontinuity
S_f	Numerical solution operator for the system of conservation laws
S_ψ	Numerical solution operator for the system of ordinary differential equation
S_{exact}	Exact wave speed
$S_{numerical}$	Numerical wave speed
t	time
t_f	Final time at which solution is required
T	Temperature
TE	Truncation error
TE_x	Truncation error in spatial discretization
Y_i	Mass fraction of ith species
X_i	Mole fraction of ith species
W_i	Molecular weight of ith species
W	Molecular weight of a mixture
$\hat{W}_n$	Iteration matrix
$\tilde{W}$	Solution of variational equation
$\mathbf{u}$	Vector of dependent variables

u_L,u_R	Left and right state of the discontinuity
u	Velocity along x-direction
$\hat{U}$	Upper triangular matrix
$\mathbf{U}(t_n)$	The numerical approximation by the spatial discretization at time t_n
U^l, U^l	Left and right upwinded values
v	Velocity along y-direction
$\mathbf{V}(t_n)$	The numerical approximation generated by the time integrator at time t_n
$\mathbf{V}^c(t_n)$	The vector of corrected solution values at time t_n
$\mathbf{V}^p(t_n)$	The vector of predicted solution values at time t_n
$\tilde{\mathbf{V}}(t_n)$	Highly accurate solution
$\mathbf{V}^*(t_n)$	An approximation solution values when advection has been treated explicitly at time t_n
$\hat{\mathbf{V}}(t_n)$	A vector of numerical solution generated by the IMEX scheme at time t_n
$V^{(i)p}(t_n)$	ith predicted derivative
$\hat{V}_c$	Roe-avergae of the conserved variables
w_i	Concentration of ith species
$\mathbf{y}(t_n)$	Local solution at time t_n
Greek	
α	Model the advection term by Fourier analysis
$\hat{\alpha}$	Parameter appearing in NDF2 method
α_i	BDF2 method coefficient
α^*	Correction vector when BDF2 method is used
β	Model the diffusion term by Fourier analysis
β^*	Correction vector when NDF2 method is used
$\tilde{\beta}$	Temperature exponent in the rate constant
η	Solar zenith angle
γ	Ratio of specific heats
$\tilde{\gamma}$	Co-efficient appearing in BDF2 method
$\tilde{\gamma}^*$	Co-efficient appearing in NDF2 method
ι	Unit complex number

Λ	Diagonal matrix of the eigenvalues of the Jacobian matrix
Λ^+	Diagonal matrix having positive element of Λ
Λ^-	Diagonal matrix having negative element of Λ
λ_i	ith eigenvalue
ν	Roe-average operator
κ	Rate constant of the ith reaction
Ω	2D computational domain
ρ	Desnsity
$\Re^m$	Real-valued vector of m component
$\tilde{\rho}$	Convergence rate
θ	A parameter used in theta method
$\hat{\theta}$	Phase angle
τ_{xx}, τ_{xy}, τ_{yy}	Components of the viscous stress tensor
$\triangle p$	Pressure jump
$\triangle x$	Mesh size
$\triangle t$	Step size
$\triangle t_{errbal}$	Step size with error balancing approach
μ	Source term factor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1917, the British Scientist, L.F. Richardson made the first attempt to predict the weather by attempting to solve partial differential equations by hand [84]. It is supposed that this was first step towards the beginnings of Computational Fluid dynamics (CFD), an important part of scientific computing.

The increasing availability, power and sophistication of computer software and hardware have led to significant growth in the field of computational fluid dynamics in a variety of fields including airplane design, car design, ship design, studies of blood flow, oil recovery, oceanography, meteorology, and astrophysics.

We will consider CFD in the fields of the atmospheric and combustion problems. Currently the active area of the research is the numerical approximation of PDEs with stiff non-linear source terms. Such problems come from the modelling of atmospheric chemistry, non-equilibrium gas dynamics, etc.

As regards atmospheric chemistry the adverse effects of pollutants have made it vital to study thoroughly their production and loss. Important examples of pollution problems include regional oxidants, acid deposition, destruction of stratospheric ozone, and the built-up of greenhouse gases, etc. It is also a fact that the harmful effect of the pollutants is not restricted to the localities of the emissions, but rather can be propagated over a widespread area. Considering the health effects of pollution, air pollution is a growing threat not only to human welfare but also other living species. The basic tools to provide detailed knowledge about the emission and dispersion include laboratory experiments, field studies and modelling analysis. Laboratory studies, however, are unable to explain the complex atmospheric phenomena in detail. Consequently a mathematical model that allows multiple processes to occur simultaneously is required for data analysis and scientific inquiry.

The governing equations for such models are non-linear, highly coupled and extremely stiff (see for example [94]). So, the complexity and nonlinearity of such models in general exclude the possibility of having an analytical solution, even in extremely simplified cases. Therefore, numerical methods are the only feasible alternative to meet the requirements for the simulation. This process divides time and space into discrete intervals, and then defines discrete variables that approximate the continuous functions, and we end up with the equations in a form suitable for numerical computation. We expect the corresponding numerical solution to converge and become a better representation of the continuous fluid.

A difficult problem in the computational solution of such problems is that of ensuring that the numerical solution does not have unphysical oscillations.

It has been accepted for a long time that upwind differencing can eliminate oscillations in the neighbourhood of shock wave at the expense of low accuracy. On the other hand, central difference schemes produce a good solution in smooth region, but are prone to oscillations in the neighbourhood of shock waves. These oscillations can be suppressed only by the introduction of an additional dissipative term (see for example [44]). Also upwind differencing can be designed to have the total variation diminishing (TVD) property in one dimension, which suppresses the spurious oscillations [36]. The only drawback is that they require specifying a complete eigensystem for the problem. In practice, this can involve considerable analytical work as well as some complications when the eigensystem lacks uniqueness [30].

The solution to one-dimensional Riemann problem describes the evolution of a single planar discontinuity separating two different but uniform fluid regions. This situation arises while solving problems related to Euler equations of gas dynamics if we consider only inviscid flow. So the solutions may consist of shocks, contact discontinuities and rarefactions. In a shock, the density, energy, pressure, and velocity are all discontinuous. At a contact discontinuity, the density and energy are discontinuous, but the pressure and velocity are continuous. At these discontinuities, conservation laws give analytical jump conditions that can be used to determine how the discontinuities evolve in time. Godunov first incorporated this analytical solution to the finite-difference method to improve accuracy at discontinuities (see [35]). His basic idea was to solve a Riemann problem cell-by-cell or region-by-region in the flow, and then to piece these local analytic solutions together.

Using this approach an upwind code comprises an interpolation procedure coupled with an approximate Riemann solver (ARS). The primary role of the ARS is to evaluate the flux at the interface given the states to the left and right of the interface.

In nonequilibrium gas dynamics, chemical reactions between the constituent gases must be modelled along with the fluid dynamics. While solving such reacting flow problems numerically, new difficulties arise that are absent in non-reacting flows. Aside from the increase in the number of equations, the main difficulties stem from the possible "stiffness"of the reaction terms and spurious oscillations that have been reported if insufficient spatial and time resolution has been used (see for example [19, 52]). Moreover the transport variables may present such strong gradients, as to practically make results irresolvable on any mesh of reasonable size. Hence such cases require a careful treatment of the advection terms. This is due to the fact that such flows have a strong non-linear component that is mainly driven by advection coupled with propagation of the species concentration.

A variety of efficient numerical schemes for hyperbolic systems of conservation laws have been developed in the recent past. These schemes evolved following the understanding of fundamental concepts from the theory of non-linear hyperbolic PDE's, such as characteristic surfaces, existence, uniqueness, and solution of the Riemann problem, etc, (see, for example, Courant and Hilbert [22], Lax[50, 51], any Yee[93]). Higher-order schemes, such as the ENO (essentially non-oscillatory) schemes, (Harten et al. [37]), the MUSCL scheme (van Leer [85]), the PPM scheme (Collela and Woodward [20]) and Roe's approximate Riemann solver [65] can be viewed as extensions of Godunov's original scheme to second-order accuracy and the method of Osher [59] is widely used. Recently Donat and Marquina [25] have devised an improved flux formula, and discussed the failure of Roe's approach on some problems.

The aforementioned schemes have been developed by making use of the theory of characteristics for the system of hyperbolic PDEs in one space dimension. They employ the characteristic decomposition of an equation into a set of scalar fields, at computational cell to evaluate the flux term at the cell interface. A discontinuous solution can be computed by supplying the characteristic equation with the appropriate jump relation. van Leer et al. [87] compared the earlier methods for the Euler and Navier-Stoke equations and suggested the use of Osher [60, 26] and

Roe's [65] schemes for their ability to accurately represents flow phenomenon such as shocks, and contact discontinuities.

With the increasing interest in high-temperature and chemically reacting flows, these methods have recently been extended to real gases by many researchers, see for example [18, 53]. Collela and Glaz [18] presented a numerical scheme for obtaining the flux from the exact solution of the Riemann problem for a real gas, that is with a non-ideal equation of state(EOS). Vinokur and collaborators (see [76]) have produced a sequence of papers on the extension of these formula to real gases, in which both the analysis of the numerical problems and the formulas produced become more and more sophisticated. Glaister [33] has presented an elegant extension of Roe's "Approximate Riemann Solver", while Liou et al. [53] present different extensions to these formulas backed by a careful analysis.

A very natural wish is to extend to non-equilibrium chemistry, which means that the concentration of the concerned species depend not only on the transport of the fluid, but also on the progress of chemical reactions, which implies that a priori determination of an EOS is not possible, and the EOS has to be constructed along with the solution process.

Just as for gases in equilibrium, the literature on numerical flux function for non-equilibrium gases is rapidly expanding. Many solutions have been presented to generalise Roe's solver (for example [2, 76]) as well as Osher's solver[1, 79] and references therein. More recently Fedkiw et al. [29] have produced the results for thermally perfect gas flows with chemistry with alternate route (ENO schemes). In all these cases very complex analysis is involved, and it may be impossible to derive simple enough expressions which can lead to an efficient calculation of fluxes, hence we have extended the Donat and Marquina approach [25] to include non-equilibrium chemistry.

The next task is to use a suitable time integration algorithm for solving ODEs arising from PDEs with the implementation of the method of lines. While choosing the time integration algorithm it is important that it has the property of keeping the time step as large as possible without sacrifying accuracy. Full details of the time integration algorithm can be found in Berzins et al.[6, 8, 9]. Of the many methods that may be used two are: the theta method (see Berzins and Furzeland [9]) and the Gear backward differentiation (BDF) formula up to order 5 (see Berzins [6]). In order to deal with the chemical kinetics arising from the atmospheric chemistry we have used the extended stability region formula of Klopfenstein [47] and have achieved very promising results in combination with Gauss Seidel iteration.

The traditional approach in the time integration, while solving the ODEs is to either control the CFL number ([84]) or the local error per step. We have controlled the local error per unit step based upon the error balancing approach ([7, 49]), which will be described in Chapter 4.

1.1 Mathematical Framework

Computational models describing the chemical transformations and transport of species have an essential role in understanding the complex processes which lead to the formation of pollutants such as greenhouse gases, acid rain and photochemical oxidants. An accurate and detailed description of the distribution of pollutants concentrations is needed over large spatial regions in order to compare with field measurements calculations. It is necessary to understand the mixing between plumes generated from the concentrated source and distributed urban and biogenic emissions, which is a difficult task because there are many processes that affect the fate of the plumes including reaction, deposition and transport (see for example [82])

The general form of the atmospheric dispersion equation in Cartesian co-ordinates can be written as (for detail see [88])

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{P}^{*}{}_{1}(\mathbf{u}, x, y, z, t) + \cdots + \mathbf{P}^{*}{}_{M^*}(\mathbf{u}, x, y, z, t), \tag{1.1}$$

where the vector $\mathbf{u}$ is the concentration of the species being considered, x, y, z are the Cartesian Co-ordinate and t is the time [88]. The function $\mathbf{P}^*_i$ $i = 1, \cdots, \mathrm{M}^*$ represents the physical and chemical process that are to be modelled. They may consist of advection, diffusion, dry deposition, wet deposition, fumigation, emission and chemical reaction (see [88]). The full equation of the model described in this thesis (see section 2.4) clearly is of the form (1.1), and for simplicity we have confined ourselves to advection and chemistry of the concerned species in modelling atmospheric flows.

In nonequilibrium gas dynamics, the chemical reactions between the constituent gases must be modelled along with the fluid dynamics. The coupled system of this form also arises in combustion problems. In contrary to the simple (to some extent) atmospheric dispersion equation (1.1), the governing equations are either the Euler equations or Navier Stoke equations if we consider viscous effects, which have been modified to include multiple gas species and appropriate chemical reactions. The discontinuity presented in the initial data will break up into a combination of shocks, rarefaction and contacts as the evolution proceeds in time (see for example [58, 84] and Chapter 2), which is not the case with atmospheric dispersion problems.

Hence restricting our attention to inviscid flow, we have essentially the Euler Equations of gas dynamics coupled with source terms representing the chemistry. In two space dimensions these equation take the form

$$\mathbf{u}_t \;+\; \mathbf{f}(\mathbf{u})_{\mathbf{x}} \;+\; \mathbf{g}(\mathbf{u})_{\mathbf{y}} \;=\; \psi(\mathbf{u}), \tag{1.2}$$

where $\mathbf{u}$ is the vector of dependant variables including momentum, energy, density and concentration for each species in the reacting mixture. The flux functions $\mathbf{f}$ and $\mathbf{g}$ describe the fluid dynamics as in the Euler Equations, while the source term $\psi(\mathbf{u})$ arises from the chemistry of the reacting species (see for example [28, 83]).

The next task is to find the solution of such complex system of partial differential equations. The most popular techniques include the method of lines and operator splitting techniques. In this thesis no attention has been paid to operator splitting, because we have concentrated on the method of lines. When operator splitting is being implemented, instead of integrating the equations(1.1) and (1.2) at once, the integration is done for each process separately. This implies that following sequence of differential equations is solved over the time interval $[t_0, t_1]$ for the atmospheric equation

$$\begin{cases} \frac{\partial \mathbf{u}_i}{\partial t} \;=\; \mathbf{P}^*{}_i(\mathbf{u}, x, y, z, t), \quad \text{for } i \;=\; 1, \cdots, \mathrm{M}^* \\ \mathbf{u}_i(x, y, z, t_0) = \mathbf{u}_{i-1}(x, y, z, t_1), \end{cases} \tag{1.3}$$

with $\mathbf{u}_0(x, y, z, t_1) \;=\; \mathbf{u}(x, y, z, t_0)$ and each $\mathbf{P}^*{}_i(\mathbf{u}, x, y, z, t)$ represents the physical and chemical process that is to be modelled, and M^* is the total number of processes being modelled. With this technique the solution of equation (1.1) at time t_1, the results of the last step in equation (1.3) is taken, i.e. $\mathbf{u}(x, y, z, t_1) \;=\; \mathbf{u}_{M^*}(x, y, z, t_1)$, and the error made in the approximation $\mathbf{u}(x, y, z, t_1)$ is first order in time. If in the next step from t_1 to t_2, the order of the processes have been reversed, the error in the solution at t_2 is second order in time. This is called Strang splitting and is commonly applied for the atmospheric problem (see [88] and references therein), and similar procedures can be applied to the combustion problem (see [28]).

The advantage of the operator splitting is that the chemistry is treated separately, hence the most efficient numerical technique can be used.

The disadvantage of operator splitting is the splitting error. In reactive flow problems low level accuracy often suffices, however, which may justify he application of operator splitting in such cases.

As explained in [28], the Strang splitting technique does not always work well in case of combustion problems. The reason is, one cannot split apart the two spatial convection terms aparts of the 2D Euler equations, beacuse the truncation error due to noncommutivity of operators causes a 'blow-up' of the solution (see for example McRae et al.[57]). This technique has worked well in case of Fedkiw [28], as the source terms are not overly stiff, and for the very stiff source term, the temperature minimizing procedure has been described.

The method of lines is frequently used to solve the set of time-dependent, non-linear coupled partial differential equations. To develop a method of lines model, the system of conservation equations and its associated boundary conditions are first discretized. This usually takes the form of finite difference formulae of higher than second order and the resulting system of differential equations is given by

$$\dot{\mathbf{U}} \;=\; \mathbf{F}_N(t, \mathbf{U}(t)), \qquad \mathbf{U}(0) \quad \text{given}, \tag{1.4}$$

where the vector $\mathbf{U}$ represents the species concentration in case of the atmospheric problem and density, momentum, energy and species concentration in case of the combustion problem.

A suitable algorithm is then chosen to integrate the resulting coupled ordinary differential equation in time, hence existing packages for ODEs can be used. Historically Liskovets [54] introduced this techniques and then Bledjian [13] applied it to the structure of laminar flame. The method of lines has been modified by Galant [32] whose uses an implicit multi-step method to integrate the resulting stiff ODEs in time. His problem was related to the ozone-oxygen flame system and uses the Gear integration method.

In order to solve the chemical equations with sufficient accuracy, a number of time steps with an implicit or semi-implicit solver is required (see Chapter 3). The advection can be computed using explicit integration techniques. This implies that when advection and chemistry are solved without operator splitting, a (semi) implicit method would be needed to solve this process in a coupled way because of the chemistry. Hence with the method of lines a prohibitedly large system of nonlinear equations has to be solved, which is computationally very expensive and hence unattractive, because more time steps are taken than necessary for advection alone. Hence operator splitting provides a way to limit the total computation needed.

1.2 The Present Work

The main aim of the present work is to present a possible global approach to accurately solving the reacting flow problems arising in both atmospheric and combustion modelling. This goal has been achieved by using the new approach i.e. controlling the local error per unit step (LEPUS) rather than the classical approach of controlling the local error per step (LEPS). Also the application of grid refinement is one of the numerical technique to be implemented for the reacting flow problems. The monitor function we have used is the spatial error rather than the classical monitor function i.e. spatial derivatives and cosine function,[62]. The key issues of the research are:

- Grid refinement: The technique of grid refinement offers the possibility to refine the grid dynamically in areas with large solution gradients. The refinement technique offers higher resolution where necessary. Where no refinement is needed, only computations on the coarse base grid are done. For example, a problem with a single discontinuity should ideally have a very fine mesh in the vicinity of the discontinuity, whereas a coarser mesh is adequate in the rest of the domain. On the other hand, the remeshing is quite computationally expensive and for some problems the saving made on the reduction of mesh points may be outweighed by the additional calculations required for remeshing. The efficiency of the remeshing process depends upon the choice of the monitor function and other remeshing criteria, and on the frequency of the updates. The frequency of updates should be chosen carefully so that the mesh keeps up with the evolving solution while avoiding unnecessary updates.

  As regards the monitor function we have used the spatial error rather than employ the existing technique of using the spatial derivatives (tends to infinity around a shock) or cosine functions [62], for example.

- Error Balancing Technique: While solving PDEs the error may be decomposed into spatial discretization error and temporal error. In general it is helpful if the temporal error does not corrupt the spatial error. We have tried to balance these errors in such a way so that the temporal should not corrupt the spatial error. This approach is already working well on both hyperbolic and parabolic partial differential equation, see for details [7, 49], with no source term. We have applied this approach first to the Leveque and Yee problem [52], then to reacting flow (Dispersion and Combustion) problems. The comparison of numerical results shows that the new method is efficient and reliable. The local growth in time or the spatial error measured in this way has been used to refine the 1D grid.

- Solution Methods for Chemical Kinetics Problems: The computationally most expensive part of the reacting flow is the the numerical treatment of the Chemical Kinetics. Hence much research has been recently put into getting fast and efficient method for solving ODEs arising from chemistry. We have implemented the NDF2 (Extended Stability region BDF2) method developed by Klofenstein [47] with the Gauss Seidel iteration [90] as an efficient solver for the stiff chemistry terms.

1.3 Overview of Contents

The primary aim of this work is to present an efficient, robust and general purpose numerical solver for reacting flow problems. Chapter 2 provides the description of mathematical model of reacting flows associated with atmospheric as well as combustion problems. The governing equations of the model as well as the techniques of the discretization have been presented.

Chapter 3 outlines special purpose solvers for systems of stiff ODEs arising from chemical kinetics. The emphasis is on efficiency for modest accuracy requirements.

Chapter 4 will give a general description of the numerical solution of a model problem. Stemming from controlling the local error per step (LEPS) a novel technique of controlling local error per unit step(LEPUS) has been introduced. The solution obtained with the new technique (LEPUS) has been compared with the already existing technique (LEPS). Additionally the new technique has been successfully implemented for grid refinement for the 1D case only. Chapter 5 gives an overview of the reacting flow problems.

In Chapter 6 conclusions are drawn regarding the effectiveness of the approach adopted in this thesis.

Chapter 2

The Governing Eqns and Solution Techniques

2.1 Introduction

In this chapter the governing equations for modelling air pollution dispersion and combustion will be described. The major task of this thesis is to find a better method for the numerical solution of such chemically reacting flow problems. When an attempt is made to solve the reacting flow equations numerically, new difficulties arise that are absent in non-reacting flows.

Aside from the increase in the number of equations, the main difficulty stems from "stiffness"of the reaction terms. This "stiffness"results from a combination of coupled fast and slow chemical reactions in the same scheme [58]. Fast reactions, as is often the case with combustion, create gradients that can be too large to result resolve on a mesh of "reasonable size". Moreover such flows may have a broad spectrum of time and length scales, thus affecting the stability limits for computations.

Obviously it would be desirable to solve reactive flow equations in such a way that avoids numerical oscillation of the solution, and hence unphysical solution values. The two simple reactive flow problems studied here are a combustion problem [28] and an atmospheric dispersion problem [92, 94]. In this chapter the two problems are described together with an outline of how the method of lines is applied to solving them.

2.2 General Conservation Laws

A conservation law states that the rate of change of the total amount of substance contained in a fixed region G is equal to the flux of that substance across the boundary of G. Any continuum physical system is described by the law of conservation of mass, momentum and energy. Hence, according to the definition of the conservation law for each conserved quantity, the rate of change of the total amount in the region is given by its flux (convective or diffusive) through the region boundary, plus whatever internal sources exist. The conservation laws in integral form are then given by

$$\frac{d}{dt}\int_G \mathbf{u}dV + \int_{\partial G} \mathbf{f}(\mathbf{u})dA = \int_G \psi(\mathbf{u})dV, \tag{2.1}$$

where the vector $\mathbf{u}$ represents the conserved quantity, $\mathbf{f}(\mathbf{u})$ is the flux vector, $\psi(\mathbf{u})$ is the source term and the volume and surface integral indicated are over the domain G and its boundary ∂G. Assuming that G is to be an infinitesimal volume and applying the divergence theorem, we get the differential form of the conservation laws

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla.\mathbf{f}(\mathbf{u}) = \psi(\mathbf{u}), \tag{2.2}$$

and in one space dimension, the differential form of conservation laws can be written

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = \psi(\mathbf{u}), \tag{2.3}$$

where

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} f(u_1) \\ f(u_2) \\ \vdots \\ f(u_m) \end{bmatrix} \quad \psi(\mathbf{u}) = \begin{bmatrix} \psi(u_1) \\ \psi(u_2) \\ \vdots \\ \psi(u_m) \end{bmatrix}.$$

This is system of m conservation laws in m unknowns u_i that depend on space x and time t. Here u_i are the dependent variables and x, t are the independent variables.

Many physical models can be described in terms of systems of such equations. For convenience, we will consider an equation in one dimension, then the differential conservation law takes the compact form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \psi(u). \tag{2.4}$$

In the case when u is a vector then bold face letters are used to indicate this.

2.3 Hyperbolic Conservation Laws

The general conservation law equation (2.3) is called the hyperbolic conservation law if the Jacobian matrix defined by (see [84])

$$J(\mathbf{u}) \;=\; \frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}, \tag{2.5}$$

has real eigenvalues $\lambda_i(\mathbf{u})$ and a complete set of linearly independent eigenvectors $\mathbf{R}^{(i)}(\mathbf{u}),\;\; i \;=\; 1, \cdots, m$ which we assume to be ordered as

$$\begin{gathered} \lambda_1(\mathbf{u}) \;<\; \lambda_2(\mathbf{u}) \;< \cdots \;< \lambda_m(\mathbf{u}), \\ \mathbf{R}^{(1)}(\mathbf{u}),\; \mathbf{R}^{(2)}(\mathbf{u}),\; \cdots, \mathbf{R}^{(m)}(\mathbf{u}). \end{gathered} \tag{2.6}$$

The above equations show that eigenvalues and eigenvectors depend on $\mathbf{u}$, and sometimes for the sake of brevity we shall omit the argument $\mathbf{u}$.

The hyperbolic conservation law is of interest because the equations of compressible fluid flow reduce to a hyperbolic system, the Euler equations, when the effects of viscosity are neglected. If we consider viscous effects also, then the governing equations are the compressible Navier-Stokes equations.

2.3.1 Characteristic Fields

Consider the hyperbolic system of m conservation laws as given by the equation (2.3) with the eigenvalues λ_i and corresponding right eigenvectors $\mathbf{R}^{(i)}$. The characteristic speed λ_i defines a characteristic field.

The characteristic field is called linearly degenerate if following identity holds

$$\nabla \lambda_i . \mathbf{R}^{(i)}(\mathbf{u}) \;=\; 0, \quad \forall \mathbf{u} \in \Re^m, \tag{2.7}$$

and in the above equation $\Re^m$ represents the set of real-valued vectors of m components and $\nabla \lambda_i$ is the gradient of the eigenvalue λ_i defined as

$$\nabla \lambda_i \;=\; \left(\frac{\partial}{\partial u_1}\lambda_i,\; \frac{\partial}{\partial u_2}\lambda_i, \cdots, \frac{\partial}{\partial u_m}\lambda_i \right). \tag{2.8}$$

The characteristic field is called a genuinely nonlinear if

$$\nabla \lambda_i . \mathbf{R}^{(i)}(\mathbf{u}) \;\neq\; 0, \quad \forall \mathbf{u} \in \Re^m, \tag{2.9}$$

and in the above equation $\Re^m$ and $\nabla\lambda_i$ have the similar meaning as defined immediately above.

In the case of the 1D Euler equations there are three characteristi fields corresponding to the three eigenvalues $\lambda^{(i)}$ $i = 1, 2, 3$. The first and the second fields are genuinely non-linear, while the second field is linearly degenerate (see for example [84]).

2.4 Atmospheric Dispersion Problems

The increasing level of air pollution makes it ever more desirable to help increase awareness and understanding of the problem. One example is that of power station plumes which are concentrated sources of NO_x emissions [41]. The photo-chemical reaction of this NO_x produced by the power station with polluted air leads to the generation of ozone at large distances downwind from the source. The transport of the plumes and chemical reactions are modelled by the atmospheric diffusion equation and in the 2D case we have the equation:

$$\begin{aligned}\frac{\partial w_i}{\partial t} &= -\frac{\partial(uw_i)}{\partial x} - \frac{\partial(vw_i)}{\partial y}\\ &+ \frac{\partial}{\partial x}\left(K_x(\frac{\partial w_i}{\partial x})\right) + \frac{\partial}{\partial y}\left(K_y(\frac{\partial w_i}{\partial y})\right)\\ &- (k_{i_1} + k_{i_2})w_i + E_i + \hat{R}_i(w_1, w_2, ..., w_{NS}), i = 1, 2, ..., NS, \qquad (2.10)\end{aligned}$$

where

- NS represents the number of species being modelled,
- w_i represents the concentration of the pollutants,
- u, v are wind velocities along the co-ordinate axes,
- K_x, K_y are diffusivity coefficients,
- k_{i_1}, k_{i_2} (i= 1, 2, ..., NS) represent the dry and wet deposition coefficients,
- E_i describes the emission sources for the ith (i =1, 2, ..., NS) compound,
- $\hat{R}_i$ represents the chemical reactions for ith compound (i= 1, 2, ..., NS).

Note the chemical source term part $\hat{R}_i$, of the above equation for $X = X(x, y)$ arises from the chemical reactions is modelled by the systems of the ordinary differential equation(ODE) (see Chapter 3). The ODE system describing the kinetic equations consists of reactions which have large variations in their time scales thus giving rise to classically stiff problems of ordinary differential equations. The numerical difficulty associated with such types of problem is that some reaction time scales will be much faster than the scales on which the solution is evolving and on which one would like to compute. This happens when the fast reactions are in near equilibrium. Stiff ODE solvers often deal with this situation by approximating true fast transients in a stable fashion using implicit methods, and then taking time steps modelling the slower transients with the required accuracy.

One example of such an ODE system is the simple chemical mechanism as given in appendix (B.0.2) which contains only 7 species and 7 coupled PDEs, but does however represents the main features of a tropospheric mechanism, namely the competition of the fast inorganic reactions with the slower reactions of volatile organic compounds. This separation in time-scales generates severe stiffness and so requires the use of an implicit stiff ODEs method.

2.4.1 Linear Advection

In Cartesian co-ordinates the dispersion of the pollutants in two space dimensions is given by

$$\frac{\partial w_i}{\partial t} = -\frac{\partial(uw_i)}{\partial x} - \frac{\partial(vw_i)}{\partial y}, \qquad i = 1, 2, \cdots, NS, \tag{2.11}$$

where w_i, u , v have the similar meaning as given by equation(2.10). Williamson (see references in [88]) has described desirable properties for advection schemes-the most important is that the scheme be positive. Negative solutions may lead to instabilities when dealing with chemical equations and may make the solution of the chemical scheme more difficult. Non-positive schemes may also lead to overshoot in the numerical solution. These arguments imply that the advection scheme needs to be positive. For this reason we have made use of positivity preserving schemes based on the van Leer limiter [7].

2.4.2 Dry Deposition

We are interested in finding the general path of the trace components from their source to their sinks. Dry deposition is one of the physical removal processes in which the species are absorbed irreversibly by oil, water or plant surfaces (see [92]). In equation (2.10) the term $k_{i_1} w_i$ (i= 1, 2, ...,NS) represents the dry deposition process and the coefficients k_{i_1} are written as (see [94])

$$k_{i_1} = \frac{\tilde{g}_{i_1}}{H_{mix}}, \tag{2.12}$$

where H_{mix} is the mixing height (a function of both space and time) and $\tilde{g}_{i_1}$ is the dry deposition velocity of the ith pollutants.

2.4.3 Wet Deposition

Another physical removal process of the trace components during their transportation is wet deposition. In contrast to dry deposition, in wet deposition the trace components are incorporated into precipitation elements (clouds, rain droplet, and aerosols). Wet deposition (incorporation of trace components into falling precipitation) ('washout') or cloud droplets('rainout') is only significant for those species that are water soluble. The factor $k_{i_2} w_i$ in equation (2.10) represents the wet deposition factor and k_{i_2} is the wet deposition coefficient, which is given as (see [94])

$$k_{i_2} = \frac{\hat{g}_i \hat{I}}{H_{mix}}, \tag{2.13}$$

where $\hat{I}$ is the precipitation intensity (measured in cm/s), H_{mix} is the mixing height (a function of both space and time; measured in cm) and $\hat{g}_i$ is the dimensionless factor represents the scavenging ratio of the ith pollutant.

2.5 Photolysis

An atmosphere is a giant photochemical reactor in which, the light source is the sun. Solar radiation not only heats planetary atmosphere, but it also drives much of the disequilibrium chemistry through photochemically initiated process. Radiation generally in the visible, and ultraviolet regions, either, fragments atmospheric constituents to produce atoms and ions or excites the constituents, without chemical change, to alter their reactivity (see for example [92]). Fragmentation of a chemical species following absorption of light is one of the most important photochemical process in atmospheric chemistry.

The photolysis rate constants, which depends on the solar zenith angle, has been written as first order rate constant in the form(see for example [82])

$$\tilde{j}_i = \tilde{a}_i exp(-\tilde{b}_i sec(\eta)), \tag{2.14}$$

where η is the solar zenith angle, $\tilde{a}_i$, $\tilde{b}_i$ are specific for each chemical reaction and i is the number of reactions. The solar zenith angle η depends upon the time of day (given by the local hour angle LHA), the time of year (given by the solar declination angle DEC) and the latitude (LAT), then the cosine of the solar angle is given by

$$\begin{aligned} cos(\eta) &= cos(LHA)cos(DEC)cos(LAT) \\ &+ sin(DEC)sin(LAT). \end{aligned} \tag{2.15}$$

Temperature-dependent rate constants have been evaluated by a standard Arrhenius expression given as

$$\kappa = AT^{\tilde{\beta}} exp(-\frac{E_a}{R_u T}), \tag{2.16}$$

where E_a is an experimental activation energy, R_u is the universal gas constant and T is the temperature is being regarded as a function of time of day.

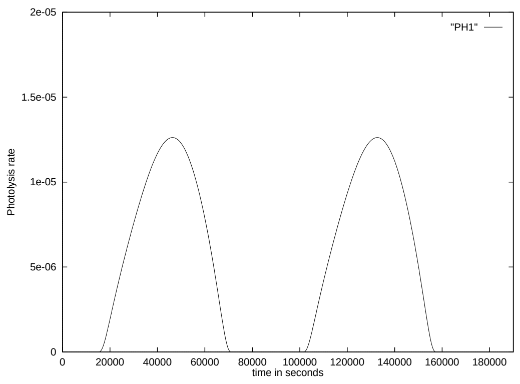
In order to have the idea of variation of photolysis rate constant with respect to time we have used simple chemical mechanism (see appendix(B.0.2)). Figure (2.1) shows the variation of photolysis rate constants in molecules cm^{-3}s^{-1} of first and third reaction for two days as a function of time and zenith angle as given by equation (2.15). For these calculations, we have taken solar declination angle (DEC) 23.27^0, the latitide has assumed 50^0 and the local hour angle is given by

$$LHA = \pi(1.0 + \frac{t}{4.32} + 4)^0, \tag{2.17}$$

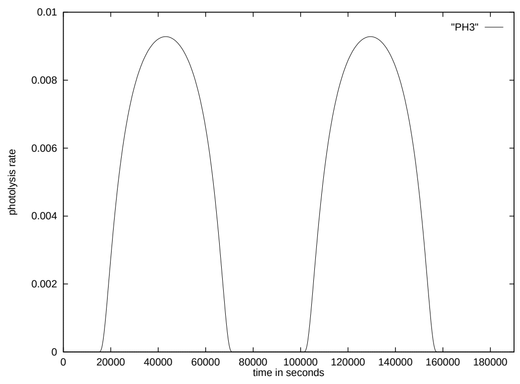
where t is time in seconds. Figure (2.1) shows the sudden rise of photolysis rate coefficient between 6AM and 6PM and the source terms will be stiff during this time.

2.6 Combustion Problems

The dynamics of compressible materials, such as gases or liquids at high pressure is governed by the Euler equations, and if we include the viscous effect then the



Variation of photolysis rate constant of first reaction



Variation of photolysis rate constant of third reaction

Figure 2.1: Variation of photolysis rate constant as a function of time and solar zenith angle in molecules cm$^{-3}s^{-1}$.

governing equations are the Navier-Stokes equations. For the computation of reactive flows in combustion devices the conservation laws of species mass, momentum and energy have to be solved and the compressible Navier-Stokes equations can be modified to include multiple gas species and appropriate chemical reactions.

The standard approach while solving the reactive flow is that the total mixture is considered as a single compressible fluid, together with the species-averaged density momentum, and energy evolving according to the conservation laws. Additionally, the mass fraction of each species is governed by separate continuity equations. These continuity equations are strongly coupled through the chemical reactions, and they are also coupled strongly to the equations for mixture via the effect of reactions on temperature and pressure [58].

These coupled equations are a set of convection equations with stiff source terms added on to model production and destruction of species in the mixture. The governing equations in vector notation and conservative form dealing with unsteady two dimensional, multi-component flows of compressible reactive gas are (see Ton et al. 1994 [83])are given as

$$\mathbf{u}_t + \mathbf{f}_x(\mathbf{u}) + \mathbf{g}(\mathbf{u})_y = \mathbf{f}_{v,x}(\mathbf{u}) \ + \ \mathbf{g}_{v,y}(\mathbf{u}) + \psi(\mathbf{u}), \tag{2.18}$$

where $\mathbf{u}$ is the vector of conserved variables, $\mathbf{f}(\mathbf{u})$, $\mathbf{g}(\mathbf{u})$ represent the invisicid parts of the fluxes, $\mathbf{f}_v(\mathbf{u})$ and $\mathbf{g}_v(\mathbf{u})_v$ are diffusive flux vectors and $\psi(\mathbf{u})$ is the source term containing species mass production (or consumption) rates. These vectors have the following forms:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho Y_1 \\ \rho Y_2 \\ \vdots \\ \rho Y_{NS-1} \end{bmatrix} \qquad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \\ \rho u Y_1 \\ \rho u Y_2 \\ \vdots \\ \rho u Y_{NS-1} \end{bmatrix} \qquad \mathbf{g}(\mathbf{u}) = \begin{bmatrix} \rho \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \\ \rho v Y_1 \\ \rho v Y_2 \\ \vdots \\ \rho v Y_{NS-1} \end{bmatrix},$$

$$\mathbf{f}_v(\mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{xx}\tau_{xy} \\ u\tau_{xx} + v\tau xy - Q_x \\ -\rho u Y_1 \\ -\rho u Y_2 \\ \vdots \\ -\rho u Y_{NS-1} \end{bmatrix} \quad \mathbf{g}_v(\mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{xx}\tau_{xy} \\ u\tau_{xx} + v\tau xy - Q_y \\ -\rho v Y_1 \\ -\rho v Y_2 \\ \vdots \\ -\rho v Y_{NS-1} \end{bmatrix} \quad \psi(\mathbf{u}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dot{w}_1 \\ \dot{w}_2 \\ \vdots \\ \dot{w}_{NS-1} \end{bmatrix}.$$

Where ρ, u and v represent the mixture density, components of mixture velocity, E is the total energy per unit volume, p is the mixture pressure, Y_i is the mass fraction of ith species, τ_{xx}, τ_{xy} and τ_{yy} are the components of the viscous stress tensor, Q_x and Q_y are the heat flux vectors, $\dot{w}_i$ stands for the mass rate of production of ith species, and finally NS represents the number of species in the mixture.

The first four equations describe the convective transport of mass, momentum and energy in a gas in two spatial dimensions. The other equations are the species continuity equations and the source terms are due to the chemical reactions. On the other hand the chemical reactions cause an abrupt change in the temperature during combustion, hence it is essential to include accurately the temperature dependencies in the equations of state used for the gas species. The most realistic model that includes the temperature dependencies is that of a thermally perfect gas, and for which the heat capacity can be a general function of temperature. Another model is the caloric perfect gas where heat capacity is constant for each species and is a function of the mass fraction (see for example [28]).

While solving the combustion problem, the hydrogen-oxygen system is an attractive object of study because its detailed reaction mechanism is well understood (in contrast, for example, with hydrocarbon oxidation), because it is the simplest realistic combustion system, and because of its potential as a fuel. The present problem consists of the detailed 37-step chemical reaction scheme relating eight species (H_2, O_2, O, OH, H, HO_2, H_2O_2 and H_2O) (see for example [28]).

2.6.1 Linear Advection

The conserved quantities represented by the vector **u** and (mass, momentum, energy and species concentration) are transported by convective and/or diffusive fluxes. The difference between the convective and diffusive fluxes is that the diffusive flux is driven by the gradient, while the convective flux exists even in the absence of the gradient. In this section we will take into account the convective transport, ignoring the diffusion. The reason is that the convective transport requires specialised numerical treatment. The diffusive fluxes can be treated by standard numerical methods. The important physical phenomena exhibited by convective conservation are the contact discontinuity, shocks and rarefaction fans. Here we will give a brief discussion of these phenomena.

2.6.2 Contact Discontinuities

A contact discontinuity is a discontinuous jump in the mass density moved by convection through a system. This kind of jump appears at the point of contact of different materials, e.g. a contact discontinuity separates oil from water, and is modelled with the simple model equation given by

$$\rho_t \;+\; u\rho_x \;=\; 0, \tag{2.19}$$

using the step-function as initial data in the above equation. In equation (2.19) u is constant and equal to the convection velocity. As the contacts are a simple convection effect, they retain any perturbation they receive. This implies that contacts are especially sensitive to numerical methods, so spurious peturbations to the contact will tend to persist and accumulate.

In the context of the Euler equations the contact discontinuity is associated with linearly degenerate field $\mathbf{R}^{(2)}$, and across which both pressure and particle velocity are constant but the density jumps discontinuously as do variables that depend on density, such as specific internal energy, temperature, sound speed, etc (see for example [84]).

2.6.3 Shock Waves

Shock waves are small transition layers of very rapid changes of physical quantities such as pressure, density and temperature and may develop spontaneously from smooth distributions. This implies that the shock jump is self-forming and self maintaining. The simplest model equation that describes shock formation is Burgers equation [61] given, in the presence of the source term, by

$$u_t + (\frac{u^2}{2})_x = \psi(u). \tag{2.20}$$

The characteristic decomposition of the source term yields that

$$\frac{du}{dt} = \psi(u), \quad \text{along} \quad \frac{dx}{dt} = f_u. \tag{2.21}$$

The above equation looks like a convection equation such as equation (2.19) with non-constant convective speed u. This equation implies that the larger u values move faster eventually overtaking smaller value, leading to the development of a right going shock of the initial data if u is positive. The shock moves at a speed that is not simply related to the characteristic speed and the shock speed is calculated by the difference between the influx and out flux of a conserved quantity into the region.

Suppose a conserved quantity u with the conservation law as given by equation (2.4) has a step function profile with one constant value extending to the left U^l and the lower constant value to the right U^r with a single transition between these two and this jump location is moving with speed S to the right. Then the integral form of the conservation law applied to any interval containing the shock gives the relation

$$S(U^r - U^l) = f^r - f^l, \tag{2.22}$$

which may be stated as: the rate at which U appears $S(U_R - U_L)$ in the interval is given by the differences in fluxes across the interval. Hence the proper speed of the shock is directly determined by and only by conservation of U via the flux f.

In the context of the one-dimensional Euler equations, shock waves are discontinuous waves associated with the genuinely non-linear fields $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(3)}$ and all the three quantities density, velocity and pressure change across a shock wave [84].

2.6.4 Rarefaction

A rarefaction is a discontinuous jump or steep gradient in properties that dissipate as a smooth expansion, and modelled by Burgers equation with the initial data having a tendency to expand. The main numerical problem arises while modelling such kind of jump is initiating the expansion (see for example [84]).

A common example is the jump in air pressure from outside to inside a balloon which dissipates as soon as the balloon bursts and the high pressure gas inside the balloon is allowed to expand.

In case of the Euler equations the rarefaction waves are associated with $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(3)}$ characteristic fields (see equation (2.6)) and the quantities, density, velocity and pressure change across a rarefaction wave [84].

2.6.5 Mass Production

The next task is to find the mass production rate $\dot{w}$ for each species arising from a set of chemical reactions. A typical reaction is given by

$$a_1A_1 \ + \ a_2A_2 \ + \ \cdots a_{NS}A_{NS} \rightarrow b_1A_1 \ + \ b_2A_2 \ + \ \cdots \ b_{NS}A_{NS}, \tag{2.23}$$

where a_i and b_i represent the reactant and product stoichiometric co-efficient and A_i stands for the chemical symbols of the involved species. The reaction rate κ in combustion problem, associated with each chemical reaction is assumed to have the following Arrhenius temperature dependent form

$$\kappa \ = \ AT^{\tilde{\beta}}exp(-\frac{E_a}{R_uT}), \tag{2.24}$$

where A, $\tilde{\beta}$, E_a, R_u, T are the pre-exponential factor, the temperature exponent, the activation energy, the universal gas constant and the temperature respectively (for unit see appendix(C)). The equations will be stiff because of the large variation of reaction rate constant κ. It is evident from equation (2.24) that $\kappa = \kappa(T)$ is a function of the temperature. The mass production rate of the species involved in the chemical mechanism can be modelled in following system of ODE

$$\dot{w}_i = W_i(P_i - L_i) \qquad\qquad i = 1, ..., NS, \tag{2.25}$$

where W_i, $P_i \ = \ b_i\Pi_{j=1}^{NS}[A_j]^{a_j}$ and $L_i \ = a_i\Pi_{j=1}^{NS}[A_j]^{a_j}$ represent the molecular weight, production and the loss rate and $[A_j]$ is the molar concentration of the species and is given by

$$[A_j] \;=\; \frac{\rho Y_{A_j}}{W_j}. \tag{2.26}$$

For a system of chemical reaction equations, we find the mass production rate of species A_i in each equation. Then we add these together to get the total mass production rate of species A_i for the entire system. Moreover the reaction mechanism is a combination of reversible reactions and we have treated the reverse reaction in the same way as forward reaction and added it to the system. In a reactive flow model the most expensive part of the calculation is often the solution of chemical kinetics. The computational cost is directly related to the number of species, the number of reactions among them (in a minor way), and the number of spatial cells in the computational domain.

On the other hand the simulation of a complex transient like that of ignition associated with a reacting system requires the full chemistry of combustion to be represented. The kinetic system not only includes a large number of species but also includes the intermediate stages of the reaction mechanism with widely varying time scales. This means that reaction mechanism is a combination of fast and slow reactions, so the concentration of each species can grow (or decay) at different rates and some time scale will be typically faster than the scale on which the solution is evolving and on which one would like to compute, (see for example [83]).

2.7 Source Terms

In the modelling of reacting flow, currently an area of research is to develop a reliable robust numerical scheme to solve the hyperbolic system with stiff nonlinear source terms (see for example [61, 81]). Although much progress has been made in this field, still there is a large question remaining concerning the optimal treatment of the source term. The simple, but computationally challenging, Leveque and Yee problem [52] plays a major role in the designing, analysing and testing of a numerical method for wave propagation problems. Leveque and Yee [52] studied a linear convective PDE with parameter-dependent source term and found that with a single discontinuity in the initial data, seemingly stable and reasonable results can be obtained, however the discontinuity is at the wrong position (see Chapter 4)!

Griffiths et al. [78] in their subsequent work have devised a scheme for which they have proved the convergence of the numerical propagation speed to the true propagation speed and found that it can oscillate and diverge if certain monoticity conditions on the source term are violated. An alternative way to handle problems with stiff source term is to change the dependent variables so that the inhomogeneous conservation law attains the form of a homogeneous conservation law. There are a number of proposed methods for treating conservation laws with source terms, see for example Roe [66] and Glaister [34], but unfortunately none is general or straightforward. Sweby [80] utilised the technique of changing the dependent variables to obtain a homogeneous PDE (no source term), but draws the conclusion that there is a large question remaining about the optimal treatment of the source term.

2.8 Finite Difference Spatial Discretization Methods

The mathematical modelling of complex systems involves the rate of changes with respect to two or more independent variables. Generally these variables represent time, length or angle and automatically give rise to either one or more partial differential equations. The problems which involve time, t, as one of the independent variable are modelled by parabolic or hyperbolic equations. Hyperbolic equations usually come from problems where discontinuities exist in time, such as shock wave, across which there are discontinuities in speed, pressure and density. The complexity and non-linearity of such model problems generally, but not always, exclude the possibility of having an analytical solution.

The main concern here is the implementation of numerical methods for solving partial differential equations. When such a numerical method is being implemented, the basic technique is to replace the continuous problem represented by the PDEs by discrete problem. Many techniques are widely used, but perhaps the most popular are the finite difference technique, the finite volume method and the finite element methods. In the finite difference approach, the values are regarded as the point values defined at grid points, while in the finite volume approach these discrete values are regarded as the average values over the finite volume. In fluid flow problems the second approach is commonly used [68]. The finite element method will not be considered further here.

2.8.1 Finite Difference Approximations to Derivatives

For a sufficiently smooth function f(x); the Taylor Theorem states that the value of $f(x)$ at any neighbourhood point $x_0 + \triangle x$ can be found if we know $f(x_0)$ and all its derivatives at $x = x_0$ by the formula [55]

$$f(x_0 + \triangle x) = f(x_0) + \sum_j \frac{(\triangle x)^j}{j!} f^{(j)}(x_0). \tag{2.27}$$

Hence by neglecting the term of third order, $O(\triangle x)^3$, and higher we can write

$$f(x_0 + \triangle x) = f(x_0) + \triangle x f^{(1)}(x_0) + \frac{(\triangle x)^2}{2} f^{(2)}(x_0) + O(\triangle x)^3, \tag{2.28}$$

and similarly we have that

$$f(x_0 - \triangle x) = f(x_0) - \triangle x f^{(1)}(x_0) + \frac{(\triangle x)^2}{2} f^{(2)}(x_0) + O(\triangle x)^3. \tag{2.29}$$

Hence neglecting the second order term immediately leads to an approximation to the first derivative $f^{(1)}(x)$ of $f(x)$ at x_0 and that is

$$f^{(1)}(x_0) = \frac{f(x_0 + \triangle x) - f(x_0)}{\triangle x} + O(\triangle x), \tag{2.30}$$

and is called a forward finite difference approximation, because $x_0 + \triangle x$ is on the right hand-side to the point x_0, at which the derivative is to be sought.
Similarly we can obtain a backward first order approximation to the derivative as follows

$$f^{(1)}(x_0) = \frac{f(x_0) - f(x_0 - \triangle x)}{\triangle x} + O(\triangle x). \tag{2.31}$$

Now subtracting the equation (2.29) from the equation (2.28) leads to

$$f^{(1)}(x_0) = \frac{f(x_0 + \triangle x) - f(x_0 - \triangle x)}{2\triangle x} + O(\triangle x)^2, \tag{2.32}$$

and is called the central difference approximation to $f^{(1)}(x_0)$ and is second order accurate.

2.8.2 Finite Difference Approximation to a PDE

Central and upwind schemes are commonly used to discretize the PDE. In order to check the behaviour of these schemes consider the equation (2.2) in the 1D case(see for example [84]), with no source term

$$\begin{aligned} u_t + au_x &= 0, \\ u(x,0) &= u_0(x), \end{aligned} \tag{2.33}$$

subjected to the appropriate initial and boundary conditions on the x-t plane. The domain of integration is $[0, L] \times [0, t_f]$. We are interested in finding the solution of the above problem and for this we discretize the domain $[0, L]$ using N equally grid spaced points given by $\triangle x = \frac{L}{N}$. The mesh points on the x-t plane are then positioned at $(j\triangle x, n\triangle t)$ with $j = 0, \cdots N$ and $n = 0, \cdots, M$, then the discrete value of $u(x,t)$ at $(j\triangle x, n\triangle t)$ will be denoted by $U_j(t_n)$, where j refers to the space discretization and n to the time discretization is called the time level.

Then the spatial derivative with a second order central approximation is given by at a particular time t_n

$$u_x = \frac{U_{j+1}(t_n) - U_{j-1}(t_n)}{2\triangle x}. \tag{2.34}$$

Suppose that $V_j(t_n)$ be the numerical approximation generated by the time integrator at t_n and the equation (2.33) can be written as

$$\frac{V_j(t_{n+1}) - V_j(t_n)}{\triangle t} + a\frac{V_{j+1}(t_n) - V_{j-1}(t_n)}{2\triangle x} = 0, \tag{2.35}$$

where the Forward Euler method has been used for the time integration. It is evident from the above equation that the only unknown is $V_j(t_{n+1})$, because all other values at the time level are known, hence we have that

$$V_j(t_{n+1}) = V_j(t_n) - \frac{c}{2}(V_{j+1}(t_n) - V_{j-1}(t_n)), \tag{2.36}$$

here $c = a\frac{\triangle t}{\triangle x}$ is the dimensionless quantity and is called the Courant number (see [58, 84]).

But unfortunately the famous Von Neumann Stability analysis [58, 84] reveals that this scheme is unconditionally unstable. For this consider the trial solution in the following form $V_j(t_n) = \hat{A}^n exp(\iota j\tilde{\theta})$, where $\hat{A}$ is the amplitude, $\tilde{\theta} = \tilde{P}\triangle x$ is the phase angle, $\tilde{P}$ is the wave number in x-direction and $\iota = \sqrt{-1}$ is the unit complex number. Putting this trial solution in equation (2.35), simplification reveals that $\hat{A} = 1 - \iota c sin(\tilde{\theta})$. The requirement for the stability is $\| \hat{A} \| \leq 1$. Unfortunately $\| \hat{A} \| = 1 + c^2 sin^2(\tilde{\theta}) \geq 1$ hence the scheme is unstable under all circumstances.

2.8.3 The First Order Upwind Scheme

The previous section showed that the central difference scheme is unconditionally unstable. One solution to this problem is to replace the central finite difference approximation to the spatial derivative u_x by a first-order one-sided approximation. Then there are two possible choices for it, the approximation (see for example [84])

$$u_x = \frac{U_j(t_n) - U_{j-1}(t_n)}{\triangle x}, \tag{2.37}$$

and

$$u_x = \frac{U_{j+1}(t_n) - U_j(t_n)}{\triangle x}. \tag{2.38}$$

The correct choice of approximation depends upon the sign of the wave propagation speed a of the partial differential equation (2.33). This is an "upwind"discretization scheme and due to Courant et al.[23]. The key feature of the scheme is that the discretization has been performed according to the sign of the wave propagation speed in the partial differential equation (2.33). The word "upwind"means that spatial discretization has been performed according to directional flow information. This implies that for positive a in equation (2.33) the upwind means "leftside"and when a is negative , "right side"is the upwind direction. For positive a the equation (2.38) combined with the Forward Euler method (as the time integrator) can be written as

$$V_j(t_{n+1}) = V_j(t_n) - c(V_j(t_n) - V_{j-1}(t_n)), \tag{2.39}$$

where $V_j(t_n)$ is the numerical approximation generated by the time integrator at time t_n. Suppose that for positive a, we use the downwind information to perform the spatial discretization and we arrived at the following scheme

$$V_j(t_{n+1}) = V_j(t_n) - c(V_{j+1}(t_n) - V_j(t_n)), \tag{2.40}$$

again the Forward Euler method has been used as the time integrator and $V_j(t_n)$ has the similar meaning. A Von Neumann stability analysis of equation (2.39) yields that [84]

$$\| \hat{A} \|^2 = (1-c)^2 + 2c(1-c)cos(\tilde{\theta}) + c^2, \tag{2.41}$$

where c is the Courant number and the stability condition implies that $\| \hat{A} \| < 1$ and which is only possible if $0 < c < 1$. Hence the scheme is conditionally stable with the stability condition

$$0 < c < 1. \tag{2.42}$$

As explained in the previous section $c = a\frac{\triangle t}{\triangle x}$, so the Courant number depends upon the speed a, the mesh spacing $\triangle x$ and the time-step $\triangle t$. In equation (2.42) a is the wave propagation speed, $\triangle x$ is chosen on the desired accuracy, hence the restriction is on the selection of the time step $\triangle t$. Now we introduce the notation a^+ and a^- in order to formulate the upwind scheme in a unified manner

$$a^+ = \max(a, 0) = \frac{1}{2}(a + \mid a \mid), \tag{2.43}$$

and

$$a^- = \min(a, 0) = \frac{1}{2}(a - \mid a \mid), \tag{2.44}$$

and $\mid a \mid$ represents the absolute value of a. It can be easily inferred from equations (2.43) and (2.44) that $a \geq 0$ implies that $a^+ = a$ and $a^- = 0$, and $a \leq 0$ gives that $a^+ = 0$ and $a^- = a$. On the line of the notation a^+ and a^- we define the Courant number as follows

$$c^+ = \triangle t \frac{a^+}{\triangle x} \quad c^- = \triangle t \frac{a^-}{\triangle x}. \tag{2.45}$$

Using the notation c^+ and c^- we can express the first order upwind scheme in the following way

$$V_j(t_{n+1}) = V_j(t_n) - c^+(V_j(t_n) - V_{j-1}(t_n)) - c^-(V_{j+1}(t_n) - V_j(t_n)), \tag{2.46}$$

where for $a \geq 0$ the second difference term vanishes, we are left with equation (2.39) and similarly for $a \leq 0$, we have equation (2.40) and note that the Forward Euler method is the time integrator and $V_j(t_n)$ is the approximation generated by the time integrator. The stability condition for the upwind scheme (2.46) is

$$0 \leq\mid c \mid\leq 1. \tag{2.47}$$

2.9 Finite Volume Method

Finite volume discretization methods are commonly used in the field of numerical fluid dynamics and can be considered as finite difference methods applied to the differential form of the conservation laws given by equation (2.2). The integral conservation laws (see equation (2.1)) for a discrete volume can be written as

$$\frac{\partial}{\partial t}\int_{\Omega} u d\Omega \; + \; \oint_{\tilde{S}} f.d\tilde{S} \; = \; \int_{\Omega} \psi(u) du, \tag{2.48}$$

and applied to a control volume Ω_j, then the above equation is replaced by the discrete form

$$\frac{\partial}{\partial t}(U_j \Omega_j) \; + \; \sum_{sides}(f.\tilde{S}) \; = \; \psi(U_j)\Omega_j, \tag{2.49}$$

where the sum of the flux terms refers to all the external sides of the control cell Ω_j and the right hand side has been obtained with a quadrature rule.

This is the general formulation of the finite volume method and the user has to define, for a selected Ω_j, how to estimate the volume and cell face areas of the control volume Ω_j. More importantly, the user has to define the flux f on the edge of the cell. In the 1D case the above equation has the following form

$$\triangle x \dot{U}_j(t) \; + \; f_{j+\frac{1}{2}} \; - \; f_{j-\frac{1}{2}} \; = \; \triangle x \psi(U_j(t)), \tag{2.50}$$

where $U_j(t)$ denotes the approximation of $U(t)$ at x_j and $\dot{U}_j(t)$ denotes the time derivative of U_j(t); and $j \; + \frac{1}{2}$ denotes evaluation at $x \; = \; \frac{(x_j \; + x_{j+1})}{2}$, and similarly $j - \frac{1}{2}$.

For the 2D case if we consider the regular Cartesian mesh on the region $\Omega \; = \; [0,1] \times [0,1]$ then the integration of the conservation laws on the jth square gives

$$\int_{\underline{A}_j} \frac{\partial u}{\partial t} = -\int_{\underline{A}_j} \left(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}\right) d\Omega + \int_{\underline{A}_j} \psi(u) d\Omega, \tag{2.51}$$

where $\underline{A}_j$ denotes the area of the square j and Ω is the integration variable defined on $\underline{A}_j$. If we apply the divergence theorem then we have that

$$\underline{A}_j \frac{\partial U_j}{\partial t} = -\oint_{C_j} (f.n_x + g.n_y) d\tilde{S} + \int_{\underline{A}_j} \psi(u) d\Omega, \tag{2.52}$$

where the flux g is along the y-direction C_j is the circumference of jth square and $\tilde{S}$ is the integration variable along that circumference. Estimating the line integral along each edge with implementation of the one point quadrature rule along the side of the square gives

$$\frac{dU_{i,j}}{dt} + \frac{[f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}]}{\triangle x} + \frac{[g_{j+\frac{1}{2}} - g_{j-\frac{1}{2}}]}{\triangle y} = \phi(U_{i,j}(x,y)), \tag{2.53}$$

where
$x_i = (i - \frac{1}{2})\triangle x$ and $y_j = (j - \frac{1}{2})\triangle y$, $i = 1, \cdots, N$, $j = 1, \cdots, N$ and where $\triangle x$ and $\triangle y$ are constants.

Once the mesh is selected, we have to decide where to define the variables. When the variables are associated with a cell a cell-centered finite volume method is defined and if the variables are attached to the mesh points, i.e., the cell vertices, then we call this a cell vertex finite volume method.

In this thesis we have used cell-centered finite volume methods on regular mesh as described in [7] and in order to estimate fluxes f and g we have adopted similar approach as described in [7], for conviently we will consider mostly 1D equations. The approach can be easily extended to 2D on a uniform grid.

2.9.1 The Higher Order Spatial Discretizations

The first order upwind discretization is a very diffusive scheme [58], implying that a higher order approximation of the flux is needed in order to obtain the desired accuracy. The dilemma regarding the higher order upwind biased linear scheme is that even though they are more stable than pure central differencing schemes these are still prone to oscillations under some circumstances. This limit is predicted by Godunov's famous theorem [91] stating that no linear convection scheme of second-order accuracy or higher can be monotonic. The answer is to use non-linear discretizations, which adjust themselves according to the local solution to maintain monoticity.

Applying the finite volume method as described in section (2.9) the discretization of the conservation law on cell $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ gives that

$$\dot{U}_j + \frac{f_{j-\frac{1}{2}} - f_{j+\frac{1}{2}}}{\triangle x} = \psi(U_j(t)), \tag{2.54}$$

where U_j is the approximation to u at x_j and $\dot{U}_j$ denotes the time derivative of $U(x_j, t)$ and $\triangle x$ is constant. At particular time t_n, let $U_j(t_n)$ be an approximation to the average of the true solution over the cell $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, then the discretization is conserved, implies that the time variation of U over the whole domain depends on the flux at the boundaries. This means that the contribution of the flux at internal cell interfaces cancels [52].

In equation (2.54) $f_{j+\frac{1}{2}}$ and $f_{j-\frac{1}{2}}$ are approximated by an upwind differencing scheme given by, for example

$$\begin{aligned} f_{j+\frac{1}{2}} &= \hat{f}(x_{j+\frac{1}{2}}, t_n, U^l_{j+\frac{1}{2}}, U^r_{j+\frac{1}{2}}), \\ f_{j-\frac{1}{2}} &= \hat{f}(x_{j-\frac{1}{2}}, t_n, U^l_{j-\frac{1}{2}}, U^r_{j-\frac{1}{2}}). \end{aligned} \tag{2.55}$$

The next task is to estimate $U^l(t_n)$ and $U^r(t_n)$. They are estimated with the help of the standard upwind technique combined with a suitable limiter [11] and at time t_n we have that

$$U^l_{j+\frac{1}{2}}(t_n) = U_j(t_n) + \frac{1}{2}(U_j(t_n) - U_{j-1}(t_n))B(r_j, 1), \tag{2.56}$$

$$U^r_{j+\frac{1}{2}}(t_n) = U_{j+1}(t_n) - \frac{1}{2}(U_{j+2}(t_n) - U_{j+1}(t_n))B(\frac{1}{r_{j+1}}, 1), \tag{2.57}$$

where $B(.,.)$ is the limiter function (see below) and r_j is given by

$$r_j = \frac{U_{j+1}(t_n) - U_j(t_n)}{U_j(t_n) - U_{j-1}(t_n)}. \tag{2.58}$$

Roe [67] shows that on uniform grids, different limiters give rise to different spatial accuracies. Three useful choices of limiter [7] are the following:
First-order method

$$B(r_j, 1) = 0. \tag{2.59}$$

Second-order method (reverts to first-order if($r_j \leq 0$) is due to van Leer with limiter function

$$B(r_j, 1) = \frac{r_j + \mid r_j \mid}{1 + \mid r_j \mid}, \tag{2.60}$$

which is the ratio of the derivative of centered and left upwind interpolants, while third order accuracy but not monotonicity can be achieved by the Leonard limiter given by, (see for detail [7])

$$B(r_j, 1) = 0.25 + 0.75 r_j. \tag{2.61}$$

For a non-uniform grid the equation (2.54) can be written as

$$\dot{U}_j + \frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\frac{1}{2}(\triangle x_j + \triangle x_{j+1})} = \frac{\triangle x_j \psi_{j-\frac{1}{2}} + \triangle x_{j+1} \psi_{j+\frac{1}{2}}}{\triangle x_j + \triangle x_{j+1}} \tag{2.62}$$

and the modified form of the equations (2.57) and (2.56) can be written as

$$U^l_{j+\frac{1}{2}}(t_n) = U_j(t_n) + \frac{\triangle x_{j+1}}{2}\frac{(U_j(t_n) - U_{j-1}(t_n))}{\triangle x_j}B(r_j, 1), \tag{2.63}$$

$$U^r_{j+\frac{1}{2}}(t_n) = U_{j+1}(t_n) - \frac{(\triangle x)_j}{2}\frac{(U_{j+2}(t_n) - U_{j+1}(t_n))}{(\triangle x)_{j+2}}B(\frac{1}{r_{j+1}}, 1), \tag{2.64}$$

where $B(.,.)$ is the limiter function (see above) and r_j is given by

$$r_j = \frac{(U_{j+1}(t_n) - U_j(t_n))/\triangle x_{j+1}}{(U_j(t_n) - U_{j-1}(t_n))/\triangle x_j}. \tag{2.65}$$

Now consider the problem defined by equation (2.33) with the initial condition

$$u(x,0) = u_0(x) = \begin{cases} a_l & \text{if } x \leq 0.2 \\ a_r & \text{if } x > 0.2, \end{cases} \tag{2.66}$$

where ($a_l \neq a_r$) are two constant values and the initial data has discontinuity at $x = 0.2$. This type of the initial value problem is called a Riemann problem. In order to solve such problems the numerical flux is needed at every mid point of the mesh at every time step. The Riemann problem can be solved exactly (as in Godunov's method [35]), but doing so would be computationally expensive especially when nonlinear equations are involved and some iterative method is needed.

In the context of the Euler equations, the Riemann problem is a slight generalization of the shock-tube problem: two stationary gases in a tube separated by a diaphragm. The rupture of the diaphragm generates a nearly centred wave system that typically consists of rarefaction waves, a contact discontinuity, and a shock wave. In the Riemann problem the particle speed is allowed to be non-zero, but the structure of the solution is same as that of the shock-tube problem [84]. In the rest of the chapter the dependence of $\hat{f}$ on x and t and U^l and U^r on t will be taken as understood.

2.9.2 Approximate Riemann Solvers

Computing solutions of equation (2.33) which contain discontinuities, such as shock waves, poses stringent requirements on the numerical schemes to solve the partial differential equations. Several numerical schemes have been devised for the solution of hyperbolic conservation laws based upon the information obtained by considering a sequence of Riemann problems. Historically Godunov [35] produced a conservative extension of the first-order upwind scheme of Courant et al. [23] to non-linear systems of hyperbolic conservation laws. He supposed that the initial data could be replaced by a piecewise set of states with discontinuities at $x_{j+\frac{1}{2}}$. He then found the exact solution to this simplified problem. After some time step $\triangle t$ (less than $\triangle x$ divided by the greatest speed wavespeed found in the Riemann solutions) he replaced the exact solution by a new piecewise constant approximation.

The first major extension to this line of approach was made by van Leer [85] and raised the order of accuracy of the method from one to two. Some well known methods have been described in the previous chapter. Van Leer et al. [87] compare some of the earlier methods for Euler and Navier Stoke equations and proposed the Osher [26, 60] and Roe [65] schemes for their ability to accurately represents flow phenomenon such as shocks, rarefactions and contact discontinuities, etc. A brief description of the Osher's and Roe's is given below:

For Osher's approximate Riemann solver [26, 60] the numerical flux is given by

$$\hat{\mathbf{f}}(\mathbf{U}^l, \mathbf{U}^r) \;=\; \frac{1}{2}\Big(\mathbf{f}(\mathbf{U}^l) \;+\; \mathbf{f}(\mathbf{U}^r)\Big) - \frac{1}{2}\int_{\mathbf{U}^l}^{\mathbf{U}^r} \mid A \mid d(\mathbf{U}), \tag{2.67}$$

where $J(U) \;=\; \frac{\partial \mathbf{f}(\mathbf{U})}{\partial \mathbf{U}}$ is the Jacobian matrix, and $\mid J \mid$ is a matrix defined by

$$\mid J \mid = \; J^+ - J^-, \tag{2.68}$$

with

$$J^+ \;=\; P\Lambda^+P^{-1}, \qquad J^- \;=\; P\Lambda^-P^{-1} \;\text{ and }\; J \;=\; P\Lambda P^-. \tag{2.69}$$

In above equation Λ represents the diagonal matrix of the eigenvalues of J. Λ^+ and Λ^- are the diagonal matrix such that Λ^+ has only positive elements of Λ. Hence Λ^- has only negative value, it implies that $\Lambda \;=\; \Lambda^+ \;+\; \Lambda^-$. The rows of the matrix P^{-1} are the left eigenvectors J, while the columns of the matrix P are the right eigenvector of J. The integral in equation (2.67) is evaluated along a path piecewise parallel to the eigenvectors of $J(\mathbf{U})$, it means that along the wave paths in the phase space of u (see for example [60]).

The Roe numerical flux is given by

$$\hat{\mathbf{f}}(\mathbf{U}^l, \mathbf{U}^r) \;=\; \frac{1}{2}\Big(\mathbf{f}(\mathbf{U}^l) \;+\; \mathbf{f}(\mathbf{U}^r)\Big) \;-\; \frac{1}{2} \mid \tilde{J}(\mathbf{U}^l, \mathbf{U}^r) \mid (\mathbf{U}^r \;-\; \mathbf{U}^l), \tag{2.70}$$

where $\tilde{J}(\mathbf{U}^l, \mathbf{U}^r)$ is a linearised form of a Jacobian matrix J and satisfying

- $\tilde{J}(\mathbf{U}^l, \mathbf{U}^r)(U^r - U^l) = \mathbf{f}(\mathbf{U}^r) - \mathbf{f}(\mathbf{U}^l)$,
- $\tilde{J}(\mathbf{U}, \mathbf{U}) = \mathbf{J}(\mathbf{U})$,
- $\tilde{J}$ has real eigenvalues with linearly independent eigenvectors.

Now $\mid \tilde{J} \mid$ is given by the equations (2.68) and (2.69) and for linear J we have that $\mid \tilde{J} \mid = J$. The equation (2.70) can also be written as

$$\hat{\mathbf{f}}(\mathbf{U}^l, \mathbf{U}^r) = \frac{1}{2}\Big(\mathbf{f}(\mathbf{U}^l) + \mathbf{f}(\mathbf{U}^r)\Big) - \frac{1}{2}\sum_{j=1}^{\mathrm{NPDE}} \tilde{\alpha}_j \mid \tilde{\lambda}_j \mid \tilde{\mathbf{R}}^{(j)}, \tag{2.71}$$

where NPDE is the number of partial differential equations, $\tilde{\alpha}_j$ is the wave strength, $\tilde{\lambda}_j$ and $\tilde{\mathbf{R}}^{(j)}$ are eigenvalues and eigenvectors of matrix $\tilde{J}$ (see for example [84]). As usual, the one-dimensional scalar equation is a useful study case, because it provides the starting point for the comparison of various approximate Riemann solvers.

In [86], van Leer considers the upwind-differencing first order schemes of Godunov, Roe and Engquist and Osher(E-O)[26] solvers for the invisicid Burgers's equation. He observes that the difference between the E-O scheme and Godunov's method lies in the treatment of transonic shocks, while Roe and Godunov's schemes differ only at transonic expansion where the exact Riemann solver, used in Godunov's method (see for example [25, 84]).

Roe's method puts in a so-called expansion shock, i.e., an entropy violating discontinuity. To prevent these expansion shocks, the flux function in Roe's scheme needs to be modified. Harten and Hyman [38] introduced an intermediate state that simulates the diffusion introduced to a Godunov-type scheme by a continuous transition between the left and right states. Roe [69] describes another modification that beaks down expansion shocks.

Donat and Marquina[25] have, however, pointed out even with the implementation of above mentioned modification the Roe's solver [65] still does not give good results when the sonic rarefraction is involved. They have proposed an improved formula and in the scalar case, their flux formula is a combination of Roe's flux and Lax-Friedrichs [75] flux and is given as

$$\hat{f}(U^l, U^r) = \begin{cases} f(U^l) & \text{if} f' > 0 \text{ in } [U^l, U^r], \\ f(U^r) & \text{if} f' < 0 \text{ in } [U^l, U^r], \\ \frac{1}{2}((f(U^l) + f(U^r) - \sigma(U^r - U^l)) & \text{else}, \end{cases} \tag{2.72}$$

where

$$\sigma = \max_{U \in [U^l, U^r]} \mid f'(U) \mid, \tag{2.73}$$

and $[U^l, U^r]$ should be understood as the range of U-values that lie between U^l and U^l.

While extending their approach to systems of conservation laws, Donat and Marquina [25] have pointed out that Roe's linearization may not always be appropriate, especially when dealing with systems of conservation laws other than the Euler equations for which the "Roe mean"might not be easily computed. So they have made use of two sets of eigenvalues and eigenvectors, one coming from the left and other coming from the right, to compute the flux at a given interface. The algorithmic description of Marquina's flux formula is as follow:

Having computed the left and right states, the local characteristic variables and fluxes are evaluated

$$\begin{aligned} \tilde{\mu}_l^{\tilde{p}} &= \mathbf{L}^{(\tilde{p})}(U^l).U^l, \quad \tilde{\xi}_l^{\tilde{p}} = (\mathbf{L}^{(\tilde{p})})(\mathbf{U}^l).\mathbf{f}(U^l), \\ \tilde{\mu}_r^{\tilde{p}} &= \mathbf{L}^{(\tilde{p})}(U^r).U^r, \quad \tilde{\xi}_r^{p} = (\mathbf{L}^{(\tilde{p})})(\mathbf{U}^r).\mathbf{f}(U^r), \end{aligned} \tag{2.74}$$

for $\tilde{p} = 1, 2, \cdots, m$ and $\mathbf{L}^{(\tilde{p})}(U^l)$ and $\mathbf{L}^{(\tilde{p})}(U^r)$ represents the left eigenvectors of the Jacobian matrix.

Suppose that $\lambda_1(\mathbf{U}^l), \cdots, \lambda_m(\mathbf{U}^l)$ and $\lambda(\mathbf{U}^r), \cdots, \lambda_m(\mathbf{U}^r)$ be the corresponding the left and right eigenvalues. Then we have the algorithm [25]

```
For $j = 1, \cdots, m$,
 if $\lambda_j(\mathbf{U})$ does not change sign in $[\mathbf{U}^l, \mathbf{U}^r]$ then
     if$\lambda_j(\mathbf{U}^l) > 0$ then
         $\tilde{\xi}_+^j = \tilde{\xi}_l^j$
         $\tilde{\xi}_-^j = 0$
     else
         $\tilde{\xi}_+^j = 0$
         $\tilde{\xi}_-^j = \tilde{\xi}_r^j$
     endif
 else
     $\tilde{\sigma}_j = \max_{\mathbf{U} \in \Gamma(\mathbf{U}^l,\ \mathbf{U}^r)} \mid \lambda_j(\mathbf{U}) \mid$
     $\tilde{\xi}_+^j = 0.5(\tilde{\xi}_l^j + \tilde{\sigma}_j \tilde{\mu}_l^j)$
     $\tilde{\xi}_-^j = 0.5(\tilde{\xi}_r^j - \tilde{\sigma}_j \tilde{\mu}_r^j)$
 endif
```

where $\Gamma(\mathbf{U}^l,\ \mathbf{U}^r)$ is a curve in phase space connecting $(\mathbf{U}^l)$ and $(\mathbf{U}^r)$. Then the Marquina's flux [25] is given by

$$\hat{\mathbf{f}}(\mathbf{U}^l, \mathbf{U}^r) \ = \ \sum_{\tilde{p}=1}^{m}(\tilde{\xi}_+^p \mathbf{R}^{(\tilde{p})}(\mathbf{U}^l) \ + \ \tilde{\xi}_-^p \mathbf{R}^{(\tilde{p})}(\mathbf{U}^r)), \tag{2.75}$$

where $\mathbf{R}^{(\tilde{p})}(\mathbf{U}^l)$, $\mathbf{R}^{(\tilde{p})}(\mathbf{U}^r)$ are right eigenvectors of the Jacobian matrix.

Marquina's numerical flux is consistent, i.e.,

$$\hat{\mathbf{f}}(\mathbf{U}^l, \mathbf{U}^r) = \mathbf{f}(\mathbf{U}),$$

and, when applied to a constant coefficient one-dimensional system, Marquina's scheme is equivalent to Roe's [65] and would yield an exact solution to the Riemann problem. It is evident from equation (2.75) that Marquina's numerical flux has s flux-splitting structure with

$$\mathbf{f}(\mathbf{U}^l, \mathbf{U}^r) \ = \ \mathbf{f}^+ \ + \ \mathbf{f}^-,$$

where

$$\mathbf{f}^- \ = \ \sum_{\tilde{p}=1}^{m} \tilde{\xi}_-^{\tilde{p}} \mathbf{R}^{(\tilde{p})}(\mathbf{U}^r), \quad \mathbf{f}^+ \ = \ \sum_{\tilde{p}=1}^{m} \tilde{\xi}_+^{\tilde{p}} \mathbf{R}^{(\tilde{p})}(\mathbf{U}^l). \tag{2.76}$$

Although these methods deal with ideal gases, the literature is continuously expanding to deal with the real gases in high-temperature and chemically reacting flows, (see for example [18, 33, 53], and also the references in [76]).

With growing interest in high temperature and non-equilibrium chemistry, these methods have been successfully extended to non-equilibrium chemistry. In this regard, many extensions to the Osher[60] and Roe[65] solvers have been proposed in the literature (see [1, 76, 79], and references therein).

Shuen et al. [76] extended Roe's [65] scheme, backed by a comprehensive and complicated analysis. While extending the Roe's scheme to non-equilibrium chemistry the Roe-average operator ν is given by (see for example [76, 84])

$$\nu(f) \ = \ \frac{\tilde{a}_r f_r \ + \ \tilde{a}_l f_l}{\tilde{a}_r \ + \ \tilde{a}^r}, \qquad \tilde{a} \ = \ \rho^{\frac{1}{2}}, \tag{2.77}$$

where ρ is the density. With the help of the above equation we have that

$$\hat{\rho} \ = \ \tilde{a}_l \tilde{a}_r,$$

$$\hat{u} \ = \ \nu(u),$$

$$\hat{e} \;=\; \nu(e),$$

$$\hat{H} = \nu(H),$$

where $\hat{\rho}$, $\hat{u}$, $\hat{e}$, and $\hat{H}$ represent the Roe-average of density, velocity, specific internal energy and total enthalpy, and similarly the average value $(\hat{Y})$ of mass fraction can be calculated as

$$\hat{Y} = \nu(Y).$$

The pressure jump is then calculated by considering the pressure p as a function of the following form

$$p \;=\; p(\rho, e, Y_1, \cdots, Y_{NS-1}), \tag{2.78}$$

where ρ, e, Y_i are respectively the density, specific internal energy and mass concentration for species i. Then the pressure jump is given by (see [76]),

$$\triangle p \;=\; \hat{p}_\rho \triangle \rho \;+\; \hat{p}_e \triangle e \;+\; \sum_{j\;=\;1}^{N-1} \hat{p}_{Y_i} \triangle Y, \tag{2.79}$$

where $\hat{p}_\rho$, $\hat{p}_e$ and $\hat{p}_{Y_i}$ are derivative of pressure with respective to ρ, specific energy e and species concentration $\hat{p}_{Y_i}$ (for more detail see [76]).

Similarly the extension of the Osher solver to non-equilibrium chemistry involves a complicated analysis, and it is difficult to derive simple enough expressions which can lead to a robust calculation of fluxes. In both cases the complexity of the extended Roe ([65]) and Osher [60]) solvers increase with larger numbers of species. Recently Fedkiw et al. [29] have used very complex ENO schemes to solve combustion problems related to non-equilibrium chemistry.

Hence, it is natural to try and introduce some simplicity, so we have used the Marquina approach [25] to both deal with the non-equilibrium as well as equilibrium chemistry, and have obtained very promising results. The only complexity involved is that it needs a complete analysis of the eigenvalues and eigenvectors of the governing equations.

2.10 Method of Lines

Several methods have evolved to solve sets of time-dependent, non-linear coupled partial differential equations [58] but the two commonly used are the method of lines and the operator splitting. In this project we have used the method of lines and have avoided using operator splitting because of the extra errors that are introduced, see [3].

The method of lines focused on the idea of separating the spatial and temporal parts of the problem regardless of nonlinearity of the governing equation. It is assumed that the spatial mesh, with constant spacing $\triangle x$, is defined by

$$x_{i+1} = x_i + \triangle x, \quad i = 1, \cdots, N-1, \quad x_1 = 0,$$

and the midpoints $x_{i+\frac{1}{2}} = x_i + \frac{1}{2}\triangle x$. This mesh partitions the interval $[0,\ L]$ into N subintervals of constant length $\triangle x$. The implementation of the spatial discretization to the PDE equation (2.4) is given by equation(2.54) and can be written as the following system of ODEs (see equation 1.4)

$$\dot{\mathbf{U}} = \mathbf{F}(\mathbf{U}(t), t). \tag{2.80}$$

The solution vector $\mathbf{U(t)}$ is given by

$$U_k(t) = U_j(x_j, t) \text{ where } k = TNOE \times (j-1) + i, \tag{2.81}$$

for $j = 1, \cdots, N+1$, $i = 1, \cdots, TNOE$ and $TNOE$ (total number of equations) represents the total number of partial differential equations or ordinary differential equations at each grid points. Hence the solution vector $\mathbf{U(t)}$ is thus made up of all the PDE variables at the mesh-point x_1, all the PDE variables at the mesh-point x_2., followed by any coupled ODEs variables, hence making a total of $TNOE \times NPTS$ solution components.

The ODEs system given by the equation (2.80), along with and initial condition, $\mathbf{U}(0)$, forms an initial value problem which can be integrated in time by standard methods. Such methods are described in the next chapter. One advantage of this method is that it is possible to use the very sophisticated ODEs solvers that now exist with many features and reliable error control.

An alternative approach is to employ a time-splitting in which one alternates between solving a system of conservation laws, with no source terms, and a system of ordinary differential equations modelling the chemistry. Applying this technique to the equation (2.4), we have that

$$\mathbf{V}(t_{n+1}) = S_f(\triangle t) S_\psi(\triangle t) \mathbf{V}(t_n). \tag{2.82}$$

In the above $S_f(\triangle t)$ represents the numerical solution operator for the conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{2.83}$$

and $S_\psi(\triangle t)$ is the numerical solution operator for the ordinary differential equation

$$\frac{\partial u}{\partial t} = \psi(u). \tag{2.84}$$

This splitting procedure, however, introduces an $O(\triangle t)$ splitting error (see for example [52, 57]. To maintain second-order accuracy, the Strang splitting [28, 52] can be used, in which the solution $\mathbf{V}(t_{n+1})$ is computed from $\mathbf{V}(t_n)$ by the procedure given as

$$\mathbf{V}(t_{n+1}) = S_\psi(\frac{\triangle t}{2})S_f(\triangle t)S_\psi(\frac{\triangle t}{2})\mathbf{V}(t_n). \tag{2.85}$$

The splitting approach is frequently used to solve atmospheric reacting flow problems, but it may be less satisfactory for combustion problems, since in reality the fluid dynamics and chemistry are strongly coupled. However, there are distinct advantages to the splitting from the standpoint of algorithm design. High quality numerical methods have been developed both for system of conservation laws and for stiff systems of ordinary differential equations. By decomposing the problem into subproblems of these types, it is possible to take advantage of these method directly.

2.11 Conclusion

The governing equations of atmospheric and combustion problems have been presented and there are NS (number of species) number of equations for the atmospheric problem, while the combustion problem has $NS + 3$ equation if we consider the 2D case of the combustion. These equations are highly non-linear consisting of the complex chemistry term arising from the chemistry.

Hence it is not possible to obtain solution analytically and numerical methods must be employed. This can be achieved by discretizing the governing equations at every point of the computational domain. Throughout this work, a finite volume approach is employed to both the 1D and 2D Leveque and Yee problem [52], 1D and 2D atmospheric dispersion problem, and 1D combustion problem [28].

With the help of the method of lines approach, the governing equations have been reduced to the form of the ODE system and upwind schemes have been used for the spatial discretization. The reason is that the upwind schemes are conditionally stable while the central difference schemes are unconditionally unstable.

Also some well knows Riemann solvers dealing with ideal as well as real gases have been considered. But due to simplicity we have used the Marquina approach [25] to deal with reacting flow related to the combustion problem. The resulting ODE system, which will be stiff because of the chemistry, has been solved with the help of the theta method time integrator, and also a brief description of the splitting error has been given.

Chapter 3

The Stiff Chemistry ODE Solution Methods

3.1 Introduction

The reactive flow models (discussed in Chapter2) are computationally very expensive to solve. The computational work is often dominated by the numerical treatment of the system of the ordinary differential equation (ODEs) describing the chemical transformation. When large scale models are under consideration then we have found by numerical experiments that more than 80% or more of the total computation time is spent on solving these ODEs. The term appearing $\hat{R}_i$ in the equation (2.10) is the non-linear term associated with the chemistry and can be cast in the following ODE system

$$\dot{\mathbf{w}} \ = \ \mathbf{P}(\mathbf{w}) - L(\mathbf{w})\mathbf{w}, \quad \mathbf{w} \ = \ [w_1, \cdots, w_{NS}]^t, \tag{3.1}$$

with given initial condition $\mathbf{w}(0)$. In the above equation $\mathbf{P}(\mathbf{w})$ is a NS-vector specifying the production terms and $\mathbf{L}(\mathbf{w})$ a $NS \times NS$ diagonal matrix defining the loss rates, NS being the number of species. The reciprocal of $L_i(\mathbf{w})$ represents the characteristic reaction time for species w_i. The component $P_i(\mathbf{w})$ and $L_i(\mathbf{w})w_i$ are non-negative and represent respectively, production and loss terms [90] and their dependence of the vector $\mathbf{w}$ on space is taken to be understood.

As regards the combustion problem, the reactions mechanism can be cast in the following form of chemical kinetics system. For the ith species we have that

$$\dot{w}_i \ = \ W_i(b_i \ - \ a_i)\Pi_{j=1}^{NS}[A_j]^{a_j} \quad i \ = \ 1, \ 2, \cdots, NS, \tag{3.2}$$

where W_i is the molecular weight of ith species, a_i and b_i are the stoichiometric coefficients of reactants and products and $[A_j]$ is the molar concentration of the jth species given by [28, 83]

$$[A_j] \ = \ \frac{\rho Y_j}{W_j}, \qquad j \ = \ 1,\ 2, \cdots, NS, \tag{3.3}$$

where Y_j is the mass fraction of the jth species and ρ is the density. Recall from (Chapter 2) that these kinetic equations are modelled as either Euler equations, if no viscous effect has been considered, otherwise, Navier-Stokes equations. It is worth noting that while dealing the atmospheric dispersion problem, the rate constants are in general function of temperature, while the photolysis rate constants are the function of solar zenith angle [82]. So each $\dot{w}_i$ in equation (3.1) will be a function of solar zenith angle and temperature and is given by

$$\dot{w}_i \ = \ \dot{w}_i(T,\ \eta,\ w_1,\ w_2,\ , \cdots, w_{NS}), \tag{3.4}$$

where T is the temperature and η is the solar zenith angle. In the combustion problem rate constant depends upon the temperature [28], so each w_i in equation (3.2) is a function of the temperature, density and mass fractions, then we have that

$$\dot{w}_i \ = \ \dot{w}_i(T,\ Y_1,\ Y_2,\ , \cdots, Y_{NS}). \tag{3.5}$$

These kinetic equations model reactions with widely varying time scales. Hence the classical stiffness problem of ordinary differential equations arises. Secondly, at each time step, the solution of the chemical equations is required at all grid cells. Hence the need is for fast and efficient special-purpose solvers. On the other hand, the accuracy level required may be modest, say 1%, and higher accuracy is considered to be unnecessary because of the various other uncertainties about the physical parameters and input data. Hence it may be satisfactory to solve the chemistry part of the calculation at a low accuracy only. In this chapter some special purpose solvers are described and examined for reactive flow models (i.e. atmospheric and combustion models). The detailed description of the governing equations for atmospheric and combustion problems is given in Chapter 2.

The method of lines approach has been used to numerically integrate the governing equation of atmospheric and combustion problems. Hence with spatial discretization scheme used in [10] for the PDE described in Chapter 2 result in a system of differential equations (see equation (1.4))

$$\dot{\mathbf{U}} = \mathbf{F}_N(t, \mathbf{U}(t)), \qquad \mathbf{U}(0) \text{ given}, \tag{3.6}$$

where the vector, $\mathbf{U}(t)$, is given by $\mathbf{U}(t) = [U_1(t), \cdots, U_N(t)]^t$ is the numerical approximation to the exact solution. A method of lines approach is used to numerically integrate the equation (3.6) thus generating an approximation $\mathbf{V}(t)$ to the vector of exact PDEs solution at the mesh points $\mathbf{u}(t)$. Presently the theta method with the iterative method of [12] has been used to solve atmospheric dispersion problem [10]. Recently Verwer [90] has reported that second order backward differentiation method (BDF2) with Gauss-Seidel iteration works well in solving ODEs from atmospheric chemistry. In the present chapter we describe an alternative to both the theta method and to BDF2 by investigating whether the NDF methods of Klopfenstein [47] as suggested by [71, 74] forms a viable alternative.

In the second part of the chapter, consideration is extended to the IMEX approach of Ascher et al. [4]. The standard example that is used by Ascher [4] is of a convection-diffusion type problem in which the explicit scheme is used for the convective term and implicit scheme for the diffusion term. In a similar manner reaction-diffusion problems can also be approximated. Many authors have made analysis of IMEX schemes. For example Basdevant et al. [5] made an experimental analysis of several IMEX schemes and Varah [89] has discussed stability properties for certain second order IMEX schemes. Ascher et al. [4] have systematically analysed the performance of such schemes and pay attention to their relative performance in the context of fast multigrid algorithms. In this chapter the stability property of second-order IMEX schemes by using NDF2 as the time integrator instead of BDF2 will be discussed. Most of this chapter has appeared in abbreviated form in [3].

3.2 The Theta Method

The theta method [9], is widely used to solve initial value problems for ordinary differential equations. Among the examples of application include the method of lines treatment of partial differential equation (Hopkins [41], Berzins, Dew and Furzerland [8]), simulation of electric power system (Johson et al. [21]) and the modelling of gas transmission (see for example Chau and Dew [17]). The theta method can be implemented for both stiff and non-stiff ordinary differential equations. When taking a reasonable size time step while solving a stiff problem, an important requirements is that the convergence of the Newton type iteration must be achieved. The Newton type iteration requires the evaluation of the Jacobian matrix. On the other-hand when code is applied to non-stiff problems, a full Jacobian matrix is not needed , as a much cheaper iteration method such as functional iteration can be used. The technique related to the changes of the iteration method is discussed in [72]. Our concern here is to consider the theta method to the case of problems arising from the chemistry which are highly non-linear and stiff due to the variation of the rate constant κ.

Recall that the method of lines approach has been used to numerically integrate equation (3.6). Hence it implies that generating an approximation $\mathbf{V}(t)$ to the vector of exact PDEs solution value at the mesh point $\mathbf{u}(t)$ and application of the theta method defines the numerical solution $\mathbf{V}(t_{n+1})$ at $t_{n+1} = t_n + \triangle t$ gives

$$\mathbf{V}(t_{n+1}) = \mathbf{V}(t_n) + (1-\theta)\triangle t\dot{\mathbf{V}}(t_n) + \theta\triangle t\mathbf{F}_N(t_{n+1}, \mathbf{V}(t_{n+1})). \tag{3.7}$$

In the above equation $\mathbf{V}(t_n)$ and $\dot{\mathbf{V}}(t_n)$ represents the numerical approximation and its time derivative at time t_n. The optimal choice of θ and the other details are given in [9]. In the following section we shall briefly discuss the local error as this will be required later in the thesis.

3.3 Local Error Estimation

The complete description of local error estimates can be found in [7, 9]. The basic technique is that the local error is defined by the local solution on $[t_n, t_{n+1}]$ by solving the following ODE system

$$\dot{\mathbf{y}}_{n+1} = \mathbf{F}_N(t, \mathbf{y}_{n+1}(t)), \tag{3.8}$$

where $\mathbf{y}_{n+1}(t_n) = \mathbf{V}(t_n)$ and the local error indicator $\mathbf{le}(t_{n+1})$ is given by

$$\mathbf{le}(t_{n+1}) = \mathbf{V}(t_{n+1}) - \mathbf{y}(t_{n+1}). \tag{3.9}$$

The common approach used by the software based on method of lines is that the local error is less than the user supplied tolerance, then we have that

$$\| \mathbf{le}(t_{n+1}) \| = \hat{\beta}Tol. \tag{3.10}$$

where $\hat{\beta}$ is the suitable acceptance factor less than 1. With the help of equation (3.7) the local error can be written as (see for example [7, 9]).

$$\begin{aligned} \mathbf{le}(t_{n+1}) &= -\mathbf{y}_{n+1}(t_{n+1}) + \mathbf{V}(t_n) \\ &+ (1-\theta)\triangle t\dot{\mathbf{V}}(t_n) + \theta\triangle t\mathbf{F}_N(t_{n+1}, \mathbf{V}(t_{n+1})). \end{aligned} \tag{3.11}$$

Expanding $\mathbf{V}(t_n) = \mathbf{y}_{n+1}(t_n)$ about t_{n+1} as given by equation (3.8) and its time derivative we have that

$$\begin{aligned} \mathbf{le}(t_{n+1}) &= \theta\triangle t(\mathbf{F}_N(t_{n+1}, \mathbf{V}(t_{n+1})) - \mathbf{F}_N(t_{n+1}, \mathbf{y}_{n+1}(t_{n+1}))) \\ &- (1-\theta)\triangle t(\dot{\mathbf{y}}_{n+1}(t_{n+1}) - \dot{\mathbf{V}}(t_n)) + \frac{1}{2}\triangle t^2\mathbf{y}^{(2)}_{n+1}(t_{n+1}) \\ &- \frac{\triangle t^3}{6}\mathbf{y}^{(3)}_{n+1}(t_{n+1}) + O(\triangle t^4), \end{aligned} \tag{3.12}$$

where superscript (i) means the ith derivative of $\mathbf{y}(.)$ with respect to time. With the implementation of the mean value theorem to the term involving $\mathbf{F}_N$, the definition(3.8) and expansion of the derivatives of $\mathbf{y}_{n+1}(t_{n+1})$ about t_n, we get the following form of the above equation

$$(I - \triangle t\theta J)\mathbf{le}(t_{n+1}) \approx (\theta - \frac{1}{2})\triangle t^2\mathbf{y}^2_{n+1}(t_n) + \frac{\triangle t^3}{2}(\theta - \frac{1}{3})\mathbf{y}^{(3)}(t_n) + O(\triangle t^4), \tag{3.13}$$

where the superscript (i) has the similar meaning as defined above, and J is the Jacobian matrix given by

$$J = \frac{\partial F_N(t_n, \mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}. \tag{3.14}$$

Equation (3.13) suggests that the value of $\theta = \frac{1}{2}$ would result in the most accurate theta formula. But this hypothesis is wrong because such analysis is based upon the leading term and the choice of an optimum value of theta has been discussed in Berzins and Fuzerland [9] and Prothero and Robinson [64]. This is evident from the equation (3.13) is that it makes use of the factor $(I - \triangle t\theta J)$, which is available in its LU decomposed form in case of the stiff ODE system. The local error estimation for a fixed step size in the following form has been implemented in the code of [9, 64]:

$$\mathbf{le}(t_{n+1}) = (\theta - \frac{1}{2})\triangle_n + (\theta - \theta^2 - \frac{1}{6})(\triangle_n - \triangle_{n-1}), \tag{3.15}$$

where $\triangle_n = \triangle t_n\hat{W}_n^{-1}(\dot{\mathbf{V}}(t_n) - \dot{\mathbf{V}}(t_{n-1}))$ and $\triangle_{n-1}$ is defined similarly and

$$\hat{W}_n = (I - \triangle t_n\theta J). \tag{3.16}$$

Both these vectors are stored by the code and also used in the predictor (see [9]). In the case when functional iteration being used with a constant step size the error estimate has the following form

$$\begin{aligned}\mathbf{le}(t_{n+1}) &= (\theta - \frac{1}{2})\triangle t_n(\dot{\mathbf{V}}(t_n) - \dot{\mathbf{V}}(t_{n-1})) \\ &+ (\theta - \theta^2 - \frac{1}{6})\triangle t_n(\dot{\mathbf{V}}(t_n) - 2\dot{\mathbf{V}}(t_{n-1}) + \dot{\mathbf{V}}(t_{n-2})). \qquad (3.17)\end{aligned}$$

The time step is accepted by the integrator if $\| \mathbf{le}(t_{n+1}) \|_w < 1$ and otherwise rejected, and $\| \mathbf{le}(t_{n+1}) \|_w$ is weighted error norm.

3.4 Klopfenstein NDF Method

In 1971 Klopfenstein [47] modified the BDF methods in such a way that they have better stability properties and lower error constant in some cases. These methods are called Numerical Differentiation Formulae or NDFs and the second order method step size may be 20% larger than ordinary BDF2 methods, the details are given in [47]. Although these methods were recommended by Shampine many years ago [71] they were not implemented in general purpose software until recently [74]. The question related to the extra cost of the NDF method is that we need only one more back value, which is present in the Milne-type error estimate used by the BDF code and no extra storage is required. As described in Chapter 1 the method of lines approach has been adopted to numerically integrate the equation (3.6) thus generating an approximation $\mathbf{V}(t)$ to the vector of exact PDEs solution values at the mesh points $\mathbf{u}(t)$. For simplicity the Klopfenstein method of order 2 , NDF2 hereafter will be described by starting from the BDF2 formula given by

$$\frac{\mathbf{V}(t_{n+1}) - \sum_{i=1}^{2} \mathbf{V}(t_{n+1-i})\alpha_i}{\triangle t\tilde{\gamma}} - \mathbf{F}(t_{n+1}, \mathbf{V}(t_{n+1})) = 0. \qquad (3.18)$$

In the above equation, the coefficients α_i and $\tilde{\gamma}$ are well-known, see [6]. With the help of Nordsieck vector form of the BDF the predicted values of the solution and the first two derivatives are given by $\mathbf{V}_n^p(t_n)$,$\mathbf{V}_n^{(1)p}(t_n)$ and $\mathbf{V}_n^{(2)p}(t_n)$ are given by using the existing derivatives

$$\mathbf{V}^{(l)p}(t_{n+1}) = \sum_{i=l}^{2} \frac{\mathbf{V}^{(i)}(t_n)(\triangle t)^{i-l}}{(i-l)!}, \qquad l = 0, \cdots, 2. \qquad (3.19)$$

The correction vector given below

$$\alpha^* = \frac{\mathbf{V}(t_{n+1}) - \mathbf{V}^p(t_{n+1})}{\triangle t\tilde{\gamma}}, \qquad (3.20)$$

can be obtained by solving the system of nonlinear equations, (see for example [6]), given by

$$\mathbf{V}^{(1)p}(t_{n+1}) \ + \ \alpha^* \ - \ \mathbf{F}(t_{n+1}, \mathbf{V}^p(t_{n+1}) \ + \ \triangle t\tilde{\gamma}\alpha^*) \ = \ 0, \quad \text{where } \tilde{\gamma} \ = \ \frac{2}{3}. \tag{3.21}$$

The predictor values are then corrected by

$$\mathbf{V}^{(i)}(t_{n+1}) \ = \ \mathbf{V}^{(i)p}(t_{n+1}) \ + \ \tilde{\gamma}_i \triangle t_{n+1}^{1-i}\alpha^*, \tag{3.22}$$

where $\tilde{\gamma}_0 \ = \ \tilde{\gamma}_2 \ = \ \frac{2}{3}$ and $\tilde{\gamma}_1 \ = \ 1$. Now the NDF2 formula described in [47] may be implemented in the same Nordsieck framework by writing it as a correction to equation (3.19)

$$\frac{\mathbf{V}(t_{n+1}) \ - \ \sum_{i=1}^{2} \mathbf{V}(t_{n+1-i})\alpha_i}{\triangle t\tilde{\gamma}} - \hat{\alpha}\frac{(\mathbf{V}(t_{n+1}) - \mathbf{V}(t_{n+1})^p)}{\triangle t\tilde{\gamma}} - \mathbf{F}(t_{n+1}, \mathbf{V}(t_{n+1})) \ = \ 0, \tag{3.23}$$

where $\frac{-1}{9} \leq \hat{\alpha} \leq \frac{1}{3}$. For simplicity we write the equation (3.23) as follow

$$\mathbf{V}^{(1)p}(t_{n+1}) \ + \ \beta^* - \mathbf{F}(t_{n+1}, \mathbf{V}^p(t_{n+1}) \ + \ \triangle t\tilde{\gamma}^*\beta^*) \ = \ 0, \tag{3.24}$$

where

$$\beta^* \ = \ \frac{\mathbf{V}(t_{n+1}) \ - \ \mathbf{V}^p(t_{n+1})}{\triangle t\tilde{\gamma}^*}.$$

Hence on comparing the coefficient $\mathbf{V}(t_{n+1})$ in equation (3.23) and (3.24) it follows that $\tilde{\gamma}^* \ = \ \frac{\tilde{\gamma}}{(1-\hat{\alpha})}$ and as $\hat{\alpha} \ = \ -\frac{1}{9}$ and it implies that $\tilde{\gamma}^* \ = \ 0.9\tilde{\gamma}$. The leading term of local truncation error of the qth order NDF method is

$$TE \ = \ \hat{\alpha}\tilde{\gamma}_q \ + \ \frac{1}{q+1} \quad \text{where } \tilde{\gamma}_q \ = \ \sum_{m=1}^{q} \frac{1}{m} \tag{3.25}$$

which reduces to the leading term of the truncation error of the BDF2 for ($\hat{\alpha} \ = \ 0$). So for ($\hat{\alpha} \ = \ 0$) the leading term of local truncation error associated with BDF2 is $\frac{1}{3}$ and that of ND2 ($\hat{\alpha} \ = \ -\frac{1}{9}$) is $\frac{1}{6}$, which is still twice that of the trapezoidal rule however. This implies that the local error estimate is halved and the step at the order two is increased by the factor of about 1.26 over BDF2 for the same error.

The comparisons of BDF and NDF made by Shampine and Reichelt [74] have shown that on a range of stiff test problem the NDF code is on average about 8% faster and uses an average of about 11% fewer steps on all problems except one.

3.5 Stability Properties

Now we concentrate on obtaining the stability properties of the NDF2 method. For ($\hat{\alpha} = -\frac{1}{9}$) the fixed step NFD2 formula is given below

$$\mathbf{F}(t_{n+1}, \mathbf{V}(t_{n+1})) - \frac{1}{6\triangle t}(10\mathbf{V}(t_{n+1}) - 15\mathbf{V}(t_n) + 6\mathbf{V}(t_{n-1}) - \mathbf{V}(t_{n-2})) = 0. \quad (3.26)$$

The characteristic polynomial when the formula given by equation (3.26) is applied to the single equation of the form $\dot{V} = \lambda V$ is given by [47]

$$\lambda \triangle t = \tilde{q} = \xi + \frac{1}{2}\xi^2 - \hat{\alpha}\tilde{\gamma}_q \xi^3, \quad (3.27)$$

where $\xi = 1 - \lambda^{-1} = 1 - exp(-\iota\phi)$ and $-\pi \leq \phi \leq \pi$. According to Dahlquist [24] for a multistep method to be A-stable it must be

- implicit
- have an order equal to or less than two

and further that among all second order methods that are A-stable the trapezoidal rule has the smallest error constant. Putting the value of ξ in equation (3.27) and simplifying gives that

$$\text{Re}(\tilde{q}) = 4sin^4(\theta)\left[1 - 3\hat{\alpha} + 12\hat{\alpha}cos(\theta)\right], \quad (3.28)$$

where $\theta = \frac{\phi}{2}$ and Re($\tilde{q}$) is the real part of q and recall that the order of the method is two (for derivation see appendix (A)).

For the method to be A-stable it requires that Re($\tilde{q}$) is non-negative for all values of ϕ which is the case if and only if $-\frac{1}{9} \leq \hat{\alpha} \leq \frac{1}{3}$. Figure (3.1) shows the stability region for three different values of $\hat{\alpha} = 0, -\frac{1}{9}, -\frac{2}{9}$. Recall from the previous section that $\hat{\alpha} = 0$ represents the stability region of the BDF2 while $\hat{\alpha} = -\frac{1}{9}$ shows the stability region of the NDF2.

Then the comparison of stability region corresponding to these two values of $\hat{\alpha}$ shows that the stability region in the right half plane (where the true solution is growing) corresponding to $\hat{\alpha} = -\frac{1}{9}$ is (desirably) smaller that the BDF2 stability region corresponding to $\hat{\alpha} = 0$, see Klopfenstein [47].

In order to have idea of above points we consider the homogeneous system of ODE given by

$$\dot{Q} = MQ, \quad (3.29)$$

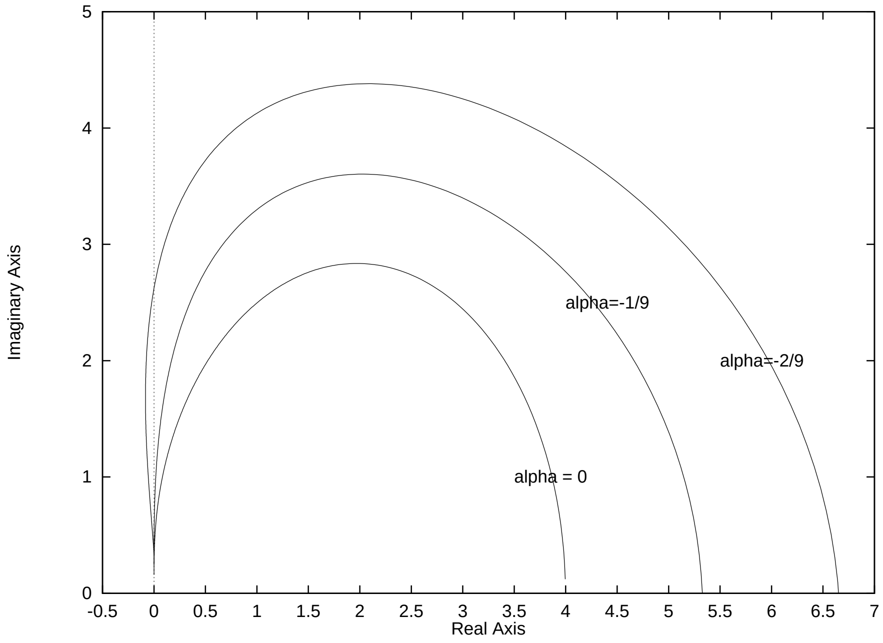


Figure 3.1: Stability region.

where M is the constant $m \times m$ matrix. It is assumed that M has distinct eigenvalues $\lambda_i, i = 1, 2, \cdots, m$. We are interested in finding the stability region of numerical method being applied to solve the above equation. For the stability of the method the requirement is that all the roots of the characteristic polynomial must be less than one. Suppose that the behaviour of the one of the root of characteristic polynomial is like this

$$r_1 \; = \; exp(\hat{\lambda}) \; + \; O(\hat{\lambda}^{q+1}), \tag{3.30}$$

where q is the order of the numerical method being used and $\hat{\lambda} = \lambda \triangle t$. It immediately follows that for small $\hat{\lambda}$ with Re $\hat{\lambda} \; > \; 0$, $\mid r_1 \mid > \; 1$ and the method is ustable. In other words, the region of stability of any convergent method cannot contain the positive real axis in the neighbourhood of the origin. Note that since the above argument is asymptotic (as $\hat{\lambda} \to 0$), we cannot conclude that the region of stability does not contain part of the positive real axis (for more detail see Lambert [48] P. 71).

3.6 Nonlinear equations splitting algorithm

When the modified Newton method is being implemented to solve the nonlinear equations at each time-step, then the system of linear equations to be solved for the (m+1)th correction to the solution $\triangle \mathbf{V}$ is

$$[I \;-\; \triangle t\tilde{\gamma} J]\,\triangle \mathbf{V}_m \;=\; \mathbf{r}(t_{n+1}^m), \tag{3.31}$$

where

$$\begin{aligned} \mathbf{J} &= \tfrac{\partial \mathbf{F}}{\partial \mathbf{V}}, \\ \mathbf{r}(t_{n+1}^m) &= -\mathbf{V}_{n+1}^m \;+\; \mathbf{z}_n \;+\; \tilde{\gamma}\triangle t\mathbf{F}(t_{n+1}, \mathbf{V}_{n+1}^m), \\ \triangle \mathbf{V}_m &= [\mathbf{V}(t_{n+1}^{m+1}) \;-\; \mathbf{V}(t_{n+1}^m)], \\ \text{and } \mathbf{z}_n &= \textstyle\sum_{i\,=\,1}^2 \mathbf{V}_{n+1-i}\alpha_i. \end{aligned} \tag{3.32}$$

The major computational task of the method of lines calculation is the solution of the system of equations (3.31). In cases when large ODEs systems result from the discretization of flow problems with complex chemistry, the CPU times may be excessive unless special iterative methods are used to solve the system of linear equations given by equation (3.31). One common approach as explained in [12] and references therein, is to take into account the ODEs function $\mathbf{F}(t, \mathbf{V}(t))$ defined by equation (3.6) and decompose it into two parts

$$\mathbf{F}(t, V(t)) \;=\; \mathbf{F}^f(t, \mathbf{V}(t)) \;+\; \mathbf{F}^s(t, \mathbf{V}(t)), \tag{3.33}$$

and in the above equation $\mathbf{F}^f(t, \mathbf{V}(t))$ represents the discretization of the advective flux term and $\mathbf{F}^s(t, V(t))$ stands for the discretization of the diffusion and source terms in the same equation. Then the nonlinear equation splitting method utilises the approximate factorisation of Jacobian matrix employed by the time integrator method within a Newton iteration

$$I - \triangle t\tilde{\gamma} J \;\approx\; [I - \triangle t\tilde{\gamma} J_f]\,[I - \triangle t\tilde{\gamma} J_s] + O(\triangle t)^2, \tag{3.34}$$

where $J_f \;=\; \frac{\partial \mathbf{F}^f}{\partial \mathbf{V}}$ and $J_s \;=\; \frac{\partial \mathbf{F}^s}{\partial \mathbf{V}}$. The technique in neglecting the advective terms J_f has been borrowed from[12] and thus in the case when no source or diffusion terms are present corresponds to using functional iteration for the advective calculation, for example see [9]. In the case when diffusion is absent or sufficiently small to be neglected in the Jacobian matrix[12], then the matrix $(I \;-\; \triangle t\tilde{\gamma} J_s)$ represents the Jacobian matrix of that part of the ODEs system corresponding to the discretization of the time derivatives and the source terms. Hence this matrix is the block-diagonal matrix with as many blocks as there are spatial elements and with each block having as many rows and columns as there are PDEs. The fact that the blocks relate only to the chemistry within each cell means that each block's equations may be solved independently.

Then the nonlinear equations splitting iteration may thus be written as

$$[I - \triangle t\tilde{\gamma} J_s]\triangle \mathbf{V}^*_m = \mathbf{r}(t^m_{n+1}), \tag{3.35}$$

and in the above equation $\triangle \mathbf{V}^*_m$ is an approximation to $\triangle \mathbf{V}_m$. The purpose of the splitting is only to speed up the solution of the nonlinear equations and providing that the iteration is continued until the residual $\mathbf{r}(\mathbf{t}^{\mathbf{m}}_{\mathbf{n+1}})$ is sufficiently small this splitting error does not have the same impact as introducing splitting at the PDE level. In order for the nonlinear equations splitting iteration defined by equation (3.35) to converge with a rate convergence r_c the necessary condition is [46] p.11 that

$$\| [I - \triangle t\tilde{\gamma} J_s]^{-1} \triangle t\tilde{\gamma} J_f \| = r_c \text{ where } r_c < 1. \tag{3.36}$$

This condition will also turn out to be important for the IMEX method considered in the next section.

Here we have concentrated on the convective transport, and ignoring diffusion. We take this simplified approach because the convective transport requires specialized numerical treatment. If present, diffusive fluxes can be treated by standard numerical methods (e.g. standard conservative central difference) [28] that are independent of those for the convective terms.

3.6.1 Gauss-Seidel Iterations

In order to find the solution of the equation (3.35), it is necessary to employ a Newton-type iterative method. Although the Newton-type iterative method is of second order of convergence, it requires the solution of a large system of linear equations which makes this method unattractive. If we have a mesh of $m_x \times m_y$ points, this results in an often prohibitively large but sparse system of $m_x n_y \times NPDE$ equations. Storage may also be a restrictive factor[55].

The great success of the implementation of the Gauss-Seidel Iterative Method for atmospheric chemistry has been reported (see [90]) and the method is now very widely used. The Jacobian matrix given by equation (3.35) can be split as follows

$$(I - \triangle t\tilde{\gamma}\hat{D} - \tilde{\gamma}\triangle t\hat{L})\triangle \mathbf{V}^*{}_{m+1} = \tilde{\gamma}\triangle t\hat{U}\mathbf{V}^*_m + \mathbf{r}(t^m_{n+1}), \tag{3.37}$$

where $\hat{L}$, $\hat{D}$, and $\hat{U}$ represent the strictly lower triangular matrix, diagonal matrix and strictly upper triangular matrix. This method completely eliminates the usage of the any kind of the decomposition and the storage requirements will also reduce considerably.

3.6.2 Convergence Test

When the Gauss-Seidel iteration is being implemented the next task is to employ a strategy to decide when to terminate the iteration during each time step. This strategy needs to be employed at each time step to avoid doing extra work and improve the efficiency. In this concern two techniques are commonly used: one is to employ a fixed number of iterations, which is unattractive because there is a risk that the code may do more work than necessary and sometimes the iterative procedure may stop before the true convergence has occurred, and the other is an adaptive procedure. We have implemented an adaptive procedure similar to that in DASSL [15]. A minimum of two iterations has been performed to evaluate the convergence rate ρ, given on the mth iteration by

$$\tilde{\rho} = \left(\frac{\| \triangle \mathbf{V}^*_{m+1} - \triangle \mathbf{V}^*_m \|_w}{\| \triangle \mathbf{V}^*_1 - \triangle \mathbf{V}^*_0 \|_w} \right)^{\frac{1}{m}}. \tag{3.38}$$

The iterations are continued until

$$\| \triangle \mathbf{V}^*_{m+1} - \triangle \mathbf{V}^*_m \|_w < ITOL, \tag{3.39}$$

where ITOL = 0.01 or 0.001 and m is the number of Gauss Seidel iterations. The fact $\tilde{\rho} > 0.95$ is taken as the convergence failure. The effect of varying ITOL is considered by Verwer [90] and in the next section.

3.6.3 Local Error

In this section we will discuss the local error being used in the NDF2 code. It is essential to find the exact behaviour of the local error, otherwise, the software may behave incorrectly and may even fail because the error estimates do not reflect the true behaviour of the error. In code based upon the BDF method the local error is being estimated by the difference between the predictor and the corrector given by (see for example, [63])

$$\mathbf{le}(t_{n+1}) = c_{n,q}(\mathbf{V}^c(t_{n+1}) - \mathbf{V}^p(t_{n+1})), \tag{3.40}$$

where $\mathbf{V}(t_{n+1})^c$ is the corrected values at the end of a step, $\mathbf{V}^p(t_{n+1})$ is the predicted value and $c_{n,q}$ represents the constant depending on the method and recent stepsize history of the integration. Sack-Davis [70] has observed that for stiff problems, the usual error estimate based upon the simple difference between the predictor and corrector overestimates the true error and has proposed the following strategy for controlling the error

$$\mathbf{le}(t_{n+1}) \;=\; \hat{W}_{n+1}^{-1} c_{n,q}(\mathbf{V}^c(t_{n+1}) \;-\; \mathbf{V}^p(t_{n+1})), \tag{3.41}$$

where $\hat{W}_{n+1}$ is the iteration matrix defined by equation (3.16) above and is available in the code in an LU decomposed form. As regards our implementation, after the successful return from the nonlinear solver then we estimate the local error. For this, first we try equation (3.40) and after calculating the weighted error norm (see later) the local error test is being made. In the case of the failure of the local error test we then used equation (3.41) to estimate the local error and repeated the same procedure. It is possible that the local error test may fail even with equation (3.41) in which case we reduce the time step. The numerical experiments have revealed that local error estimated by the equation (3.41) have given more promising results as compared to the local error estimated by the equation (3.40).

3.7 Numerical Results

We have used the three test problems to compare the performance of NDF2 Method with the theta method and the BDF2 method. The code for NDF2 method has been developed by changing a few constants in a BDF2 code. The constant $\tilde{\gamma}$ has been replaced with $\tilde{\gamma}^*$ (see Section (3.4)) and also some changes have been made to local error estimation.

The test on the standard DETEST problems [27] reveals that the predicted accuracy inprovements was achieved in practice. The experiments have revealed that NDF and BDF codes used more time-steps for atmospheric problems than the theta method code [9], but did less work per step. The aforementioned hypothesis will be tested on three test problems with the effect of using the Gauss-Seidel method.

Problem 1 This problem consists of 20 species and 25 reactions with constant reaction rates from atmospheric chemistry and is copied from [90]. The initial concentrations of the species in ppm units are given in Table (3.1) and the reaction schemes has been given in appendix (B.0.1). The problem is highly stiff because

No.	Name of the Species	Initial concentration (ppm)	No.	Name of the Species	Initial concentration (ppm)
1	NO_2	0.000	11	$C2O3$	0.000
2	NO	0.200	12	CO_2	0.000
3	O^3P	0.000	13	PAN	0.000
4	O_3	0.040	14	$CH3O$	0.000
5	HO_2	0.000	15	HNO_3	0.000
6	OH	0.000	16	O^1D	0.000
7	$HCHO$	0.100	17	SO_2	0.007
8	CO	0.30	18	SO_4	0.000
9	ALD	0.010	19	NO_3	0.000
10	$MEO2$	0.000	20	N_2O_5	0.000

Table 3.1: The initial concentration of problem 1.

for the ODE system the Lipschitz constant is about 1.5×10^7 and the simulation time is 60 minutes.

Problem 2 This problem originates from the simplified chemistry [10] and has only 7 species and 7 reactions with photolysis rate time-dependent. The photolysis rate constant has been given in equation (2.14), which shows that it depends upon the solar zenith angle, which has been given in equation (2.15). Again it is evident that solar zenith angle depends upon the time of day, the time of year and the Latitude. In our calculation we have taken the time of year (given by the solar declination angle DEC) 23.27^0, the latitude has been assumed 50^0 and the local hour angle has been evaluated by the following expression

$$LHA \ = \ \pi(1.0 \ + \ t/4.32 \ + \ 4)^0,$$

where time, t, is in seconds. The temperature dependent rate constants have been evaluated with the standard Arrhenius expression, and temperature T in degree kelvin has been evaluated according to the following expression (see for example [82])

$$T \ = \ 289.86 \ + \ 8.3sin((7.27 \times 10^{-5}t) - 1.96).$$

The initial concentrations of the species in molecules/cm^3 have been displayed in Table (3.2) and the reaction scheme have been given in appendix(B.0.2). The

No.	Name of the Species	Initial concentration (mol/cm^3)
1	NO_2	1.0×10^9
2	NO	1.0×10^9
3	O_3	1.0×10^{11}
4	ROC	1.0×10^{11}
5	RP	0.0
6	SGN	0.0
7	$SNGN$	0.0

Table 3.2: The initial concentrations of Problem 2.

simulation time is 1.8×10^5 seconds, (see [10] for details) or two days.

Problem 3 This problem with lumped chemistry was obtained from systematic reduction of the Extended Carbon Bond Mechanism CBMEx[39] and consists of 29 species and 59 reactions with non-constant reaction rates. In order to calculate the solar zenith angle and temperature, the proocedure as described in Problem 2 has been adopted. The initial concentrations in (molecules/cm^3) are displayed in Table (3.3). The full extended Carbon Bond Mechanism, see [39] with 205 reactions

No.	Name of the Species	Initial concentration (mol/cm^3)	No.	Name of the Species	Initial concentration (mol/cm^3)
1	NO_2	1.696×10^{11}	16	$MEO2$	0.000
2	NO	4.006×10^{12}	17	PAN	0.000
3	O	0.000	18	PAR	7.618×10^{12}
4	O_3	0.000	19	ROR	0.000
5	NO_3	0.000	20	KET	0.000
6	O^1D	0.000	21	OLE	3.195×10^{11}
7	H_2O	2.460×10^{17}	22	ETH	7.594×10^{11}
8	OH	0.000	23	TOL	3.072×10^{11}
9	HO_2	0.000	24	$CRES$	0.000
10	N_2O_5	0.000	25	$TO2$	0.000
11	$HONO$	0.000	26	$OPEN$	0.000
12	CO	5.680×10^{13}	27	XYL	1.954×10^{11}
13	$FORM$	2.077×10^{10}	28	$MGLY$	0.000
14	$ALD2$	1.029×10^{10}	29	$ISOP$	0.000
15	$C2O3$	0.000			

Table 3.3: The initial concentrations of Problem 3 with lumped chemistry.

(non-constant reaction rates) and 90 species was also used as a test problem but produces almost identical performance profiles to the lumped version. The initial concentrations in (molecules/cm^3) for this full chemistry are given in Table (3.4). The simulation time is 1.8×10^5 seconds again.

No.	Name of the Species	Initial concentration (mol/cm^3)	No.	Name of the Species	Initial concentration (mol/cm^3)
1	NO_2	5.410×10^{10}	46	$HTMA$	0.000
2	NO	1.600×10^{12}	47	$PNO2$	0.000
3	O	0.000	48	$DNIT$	0.000
4	O_3	0.000	49	ETH	7.600×10^{11}
5	NO_3	0.000	50	$ETO2$	0.000
6	$O1D$	0.000	51	CH_4	4.551×10^{13}
7	H_2O	2.460×10^{17}	52	OZD	0.000
8	OH	0.000	53	$ACAC$	0.000
9	HO_2	0.000	54	TOL	2.780×10^{11}
10	N_2O_5	0.000	55	$BO2$	0.000
11	HNO_3	0.000	56	$CRES$	0.000
12	$HONO$	0.000	57	TO_2	0.000
13	PNA	0.000	58	BZA	0.000
14	H_2O_2	0.000	59	$BZO2$	0.000
15	CO	5.680×10^{13}	60	$PHO2$	0.000
16	CO_2	0.000	61	$PBZN$	0.000
17	$FORM$	2.080×10^{10}	62	PHO	0.000
18	H_2	0.000	63	$NPHN$	0.000
19	$FROX$	0.000	64	CRO	0.000
20	$PROX$	0.000	65	$CRO2$	0.000
21	$FACD$	0.000	66	$NCRE$	0.000
22	$ALD2$	0.000	67	$OPEN$	0.000
23	$C2O3$	0.000	68	$ACID$	0.000
24	$MEO2$	0.000	69	XYL	1.960×10^{11}
25	PAN	0.000	70	$XLO2$	0.000
26	$MPNA$	0.000	71	$XINT$	0.000
27	MEO	0.000	72	$MGLY$	0.000
28	$MNIT$	0.000	73	$MGPX$	0.000
29	$MEN3$	0.000	74	$OPPX$	0.000
30	$MEOH$	0.000	75	$ISOP$	0.000
31	$AONE$	0.000	76	$EPOX$	0.000
32	$ANO2$	0.000	77	$ISO1$	0.000
33	PAR	7.630×10^{12}	78	$ISO2$	0.000
34	$RO2$	0.000	79	$ISO3$	0.000
35	$RO2R$	0.000	80	$ISO4$	0.000
36	X	0.000	81	$MACR$	0.000
37	NTR	0.000	82	MVK	0.000
38	ROR	0.000	83	$ISNT$	0.000
39	KET	3.150×10^{11}	84	ISN	0.000
40	D	0.000	85	$DISN$	0.000
41	$AO2$	0.000	86	$MV1$	0.000
42	OLE	3.200×10^{11}	87	$MV2$	0.000
43	$CRIG$	0.000	88	$MAC1$	0.000
44	$MCRG$	0.000	89	$MAC2$	0.000
45	$HOTA$	0.000	90	$MVNT$	0.000

Table 3.4: The initial concentrations of problem 3 with full chemistry.

3.7.1 Results Discussion

The following notations are used to present the test results

- **Step** = the number of integration steps,
- **Fun** = the number of residual evaluations,
- **TOL** = error tolerance used to define RTOL and ATOL,

- **ATOL** = absolute error tolerance $= 10^{-6}\times$ TOL for problem 1, $= 10^{5}$ for problems 2 and 3,
- **RTOL** = relative error tolerance = **TOL**,
- **ITOL** = Gauss-Seidel tolerance,
- **G-S** = number of Gauss-Seidel iterations,
- $\mathbf{SD_T}$ = the number of the significant digits for the maximum relative error at the specified time t_f, defined by

$$SD_T \ = \ -log_{10}\left(max_i\left(\frac{\mid V_i(t_n) \ - \ \tilde{V}_i(t_n)\mid}{\tilde{V}_i(t_n)}\right)\right), \tag{3.42}$$

  shows the accuracy of the calculated results, and $\tilde{V}_i(t_n)$ is the highly accurate solution and has been estimated by using DASSL[15] with much tighter tolerances.

The numerical results on Problem 1 (see Table 3.7) show that the number of Gauss-Seidel iterations per step are comparable with those in [90] albeit obtained using a somewhat different nonlinear Gauss-Seidel method without Aitken Extrapolation. The comparison of number of steps to Verwer [90] results shows that at TOL=0.1, the number of steps are almost similar and there are considerable reduction in the number of Gauss Seidel iterations, even though Aitken extrapolation technique has not been applied. For TOL=0.01, the NDF2 showed the improved performance, the code took 71 steps as compared to 132 steps for ITOL=0.01 and for ITOL=0.001, the number of steps taken by the new code are 112 as compared to 132 steps. Besides this, the Gauss-Seidel iterations have decreased considerably.

As regrads CPU time, it is an approximate value and implementation and machine dependent. The given time is an indicative for comparison purpose (on a Silicon Graphics Indigo workstation, using the Fortran77 Complier Options -g -static -mips2). We have noted the CPU time for Problem 3, which has 29 species and 59 reactions. For ITOL=0.001 and RTOL=0.01, its value is 0.08 second, which is less as compared to Verwer [90] values, even tough his Problem has 20 species and 25 reactions. A particular point to note on the results here is that when $ITOL \ = \ 0.001$ for Problem 1 then the work increases by 50%.

Full matrix	Problem 1			Problem 2			Problem 3		
	Theta	BDF2	NDF2	Theta	BDF2	NDF2	Theta	BDF2	NDF2
Steps	36	40	39	783	1016	986	744	966	978
Jac	11	14	13	48	73	67	66	74	75
Fun	354	352	337	2427	1120	1079	2372	1110	1098
SD	2.33	1.36	1.66	2.23	2.45	2.48	1.83	2.83	2.92

Table 3.5: The results of the chemical kinetics arising from atmospheric chemistry using full linear algebra for 0.1 relative tolerance.

Full matrix	Problem 1			Problem 2			Prob lem 3		
	Theta	BDF2	NDF2	Theta	BDF2	NDF2	Theta	BDF2	NDF2
Steps	59	74	64	797	1108	1042	705	985	1010
Jac	20	14	16	46	77	77	62	76	73
Fun	630	399	426	2473	1236	1169	2272	1154	1146
SD	2.53	2.88	2.68	2.42	3.37	2.34	2.47	2.97	3.06

Table 3.6: The results of the chemical kinetics arising from atmospheric chemistry using full linear algebra for 0.01 relative tolerance.

Gauss Seidel TOL	NDF2 ITOL	Problem1		Problem 2		Problem3	
		0.01	0.001	0.01	0.001	0.01	0.001
0.1	Steps	51	71	982	1015	1024	1054
	Jac	17	29	67	70	88	96
	Fun	431	715	1080	1116	1166	1210
	G-S	221	460	2171	2314	2883	3847
	SD	1.99	2.02	2.57	3.03	2.24	2.39
0.01	Steps	71	112	1037	1027	980	1002
	Jac	18	42	73	71	80	88
	Fun	476	1045	1170	1142	1163	1192
	G-S	379	647	2421	2542	3153	4218
	SD	2.71	2.84	4.4	2.89	2.92	2.39

Table 3.7: The results of the chemical kinetics arising from atmospheric chemistry using Gauss Seidel method.

A comparison between the theta and NDF2 (see Tables (3.5) and (3.6)) methods shows that NDF2 uses less function evaluations but takes more steps and Jacobian evaluations. This is probably due to the theta method's sophisticated error estimator [9] and its double or halving stepsize strategy. It is also worth noting that the theta codes error estimator requires one extra function evaluation and back solve per step, and so accounts for the much larger number of function calls. A comparison between Tables (3.5) and (3.6) shows that when the Gauss Seidel method is used there are a few more Jacobian evaluations for Problem 3. This seems to arise because of the fact that for Problem 3 19 of the 29 equations are not diagonally dominant. Analysis of the Jacobian matrices shows that species 8, the OH radical, destroys the diagonal dominance of a large part of the matrix. Table (3.7) also shows that there is significant cost penalty in terms of numbers of iteration associated with using $ITOL = 0.001$ but that there is a also increase in the accuracy.

3.8 IMplicit-EXplicit Methods

In this section we will explore the effect of the improved performance of the NDF method over BDF methods when it is applied to the IMEX approach used by Ascher et al. and categorise the relationship between the IMEX approach and the nonlinear equations splitting method described in Section (3.6). For this we write the ODE function as given by equation (3.26) in the following form

$$\mathbf{F}(t_{n+1}, \mathbf{V}(t_{n+1})) = \mathbf{F}^f(t_{n+1}, \mathbf{V}(t_{n+1})) + \mathbf{F}^s(t_{n+1}, \mathbf{V}(t_{n+1})), \tag{3.43}$$

where $\mathbf{F}^f(t_{n+1}, \mathbf{V}(_{n+1}))$ and $\mathbf{F}^s(t_{n+1}, \mathbf{V}(t_{n+1}))$ have defined in Section (3.6). Ascher et al. [4] in their approach effectively replace the non-stiff part of the ODE, i.e., $\mathbf{F}^f(t_{n+1}, \mathbf{V}(_{n+1}))$ with an explicit method to get

$$\mathbf{F}^f(t_{n+1}, \mathbf{V}(t_{n+1})) = 2\mathbf{F}^f(t_n, \mathbf{V}(t_n)) - \mathbf{F}^f(t_{n-1}, \mathbf{V}(t_{n-1})). \tag{3.44}$$

In another approach, Frank et al. [42] replaced the implicit term in non-stiff part of the ODE as given by equation (3.26) as follows:

$$\mathbf{V}^*(t_{n+1}) = 2\mathbf{V}(t_n) - \mathbf{V}(t_{n-1}). \tag{3.45}$$

We have adopted here second approach and which gives the NDF IMEX method considered here:

$$\begin{aligned} &\mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1})) + \mathbf{F}^s(t_{n+1}, \hat{\mathbf{V}}(t_{n+1})) \\ &- \frac{1}{6\triangle t}\Big(10\hat{\mathbf{V}}(t_{n+1}) - 15\mathbf{V}(t_n) + 6\mathbf{V}(t_{n-1}) - \mathbf{V}(t_{n-2})\Big) = 0, \end{aligned} \tag{3.46}$$

where $\mathbf{V}^*(t_{n+1})$ is given by equation (3.45) and $\hat{\mathbf{V}}(t_{n+1})$ is the solution value evaluated by this method at the end of the step. Now the Newton iteration is identical to that given by equation (3.31) except that the residual on the right hand side is defined by

$$\begin{aligned} \hat{\mathbf{r}}(t^p_{n+1}) &= -\hat{\mathbf{V}}^p(t_{n+1}) + \mathbf{z}_n + \tilde{\gamma}\triangle t\mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1}) \\ &+ \tilde{\gamma}\triangle t\mathbf{F}^s(t_{n+1}, \hat{\mathbf{V}}^p(t_{n+1})). \end{aligned} \quad (3.47)$$

In this case a crude approximation to the norm of the inverse iteration matrix [71] is given by the observed rate of convergence r_c^{imex}

$$\| [I - \triangle t\tilde{\gamma}J_s]^{-1} \| \approx r_c^{imex}, \quad \text{where } r_c^{imex} < 1. \quad (3.48)$$

It is worth noting that the cost of the first IMEX iteration is identical to that one iteration of the splitting method as described in Section (3.6), but thereafter the term $F^f(t_{n+1}, V^*(t_{n+1}))$ does not have to be evaluated.

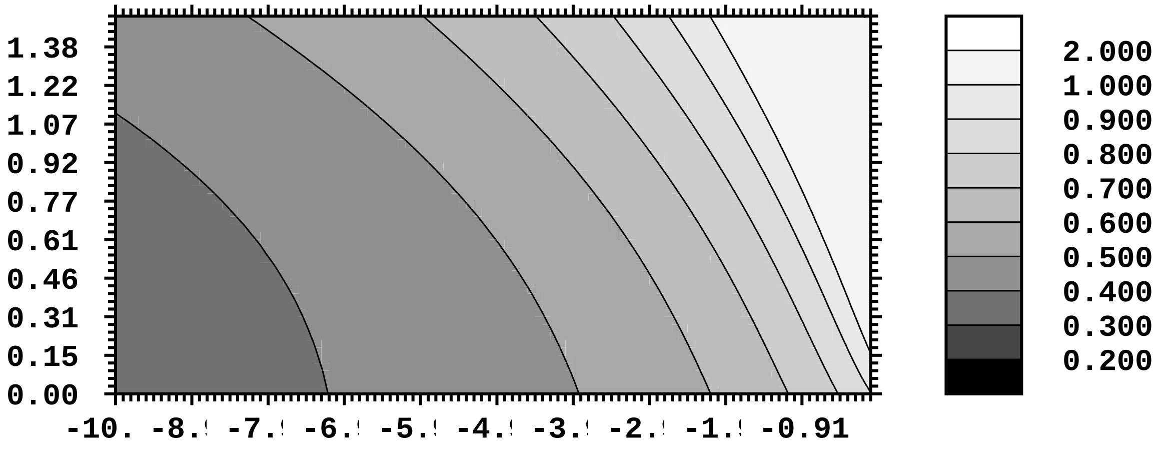


Figure 3.2: IMEX stability region.

The test equation considered by Ascher et al. [4] is given by

$$\dot{V} = (\alpha + \iota\beta)V \quad \alpha, \beta \text{ real}, \quad (3.49)$$

which has been obtained by a Fourier analysis of the advection diffusion equation. In the above equation β models the advective terms and α the diffusive term. When the BDF2 method is under consideration the characteristic polynomial ($\hat{\alpha} = 0$) is

$$\phi(z) = (\frac{3}{2} - \alpha\triangle t)z^2 - (2 + 2\iota\beta\triangle t)z + (\frac{1}{2} + \iota\beta\triangle t), \quad (3.50)$$

while with the help of NDF2 the characteristic polynomial has the following form

$$\phi(z) = (\frac{10}{6} - \alpha\triangle t)z^3 - (\frac{5}{2} + 2\iota\beta\triangle t)z^2 + (1 + \iota\beta\triangle t)z - \frac{1}{6}. \quad (3.51)$$

The stability contours of the above polynomial are displayed in Figure (3.2) in which the horizontal axis is α and the vertical axis is β. The comparison between Figure (3.2) and (5) in [4] shows that the NDF2 method is stable for purely imaginary eigenvalues, unlike BDF2. In the case when the nonlinear equations splitting method of Section (3.6) is applied to the same model equation (3.49) used by Ascher, then from equation (3.36) the convergence condition is

$$\mid [I - \triangle t\tilde{\gamma}\alpha]^{-1}\triangle t\tilde{\gamma}\iota\beta \mid < r_c, \quad (3.52)$$

and it is evident that the NDF2 method is stable for purely imaginary $\triangle t\beta$ in the range $[0, \frac{r_c}{\tilde{\gamma}\triangle t}]$, unlike the BDF2 method and so has slightly superior stability property.

3.8.1 The Extended Test Equation of Frank et al.

Frank et al. [42] have emphasised that there is a need to generalise the stability decomposition used by Ascher to model some aspects of the atmospheric diffusion equation by allowing α and β in equation (3.49) to be complex. In this concern two situations have been followed by Frank et al. In the first choice they consider the values of β for which the method is A-stable with respect to α. The second option [42] is to recognise that while A-stability is valuable: it is , in many practical situation, possible to settle for A(α)-stability. We have adopted here the second option in that β is forced to lie in the stability region of the explicit NDF2 method given by

$$F^f(t_{n+1}, V^*(t_{n+1})) - \frac{1}{6\triangle t}(10\hat{V}(t_{n+1}) - 15V(t_n) + 6V(t_{n-1}) - V(t_{n-2})) = 0, \quad (3.53)$$

and a similarly modified BDF2 method. In these two cases the stability regions are given by the interior of the semi-circular domains shown in Figure (3.3) and for NDF2 the maximum possible values of $\mid \beta\triangle t \mid \leq 1.775$ and for ≤ 1.3 for BDF2. Figure (3.4) gives the boundaries of the stability region for NDF2 ($\hat{\alpha} = -\frac{1}{9}$) and for BDF2 ($\hat{\alpha} = 0$) with β has been chosen any of the stable values in Figure(3.3). The stability regions in Figure (3.4) are the exteriors of the semi-circular regions.

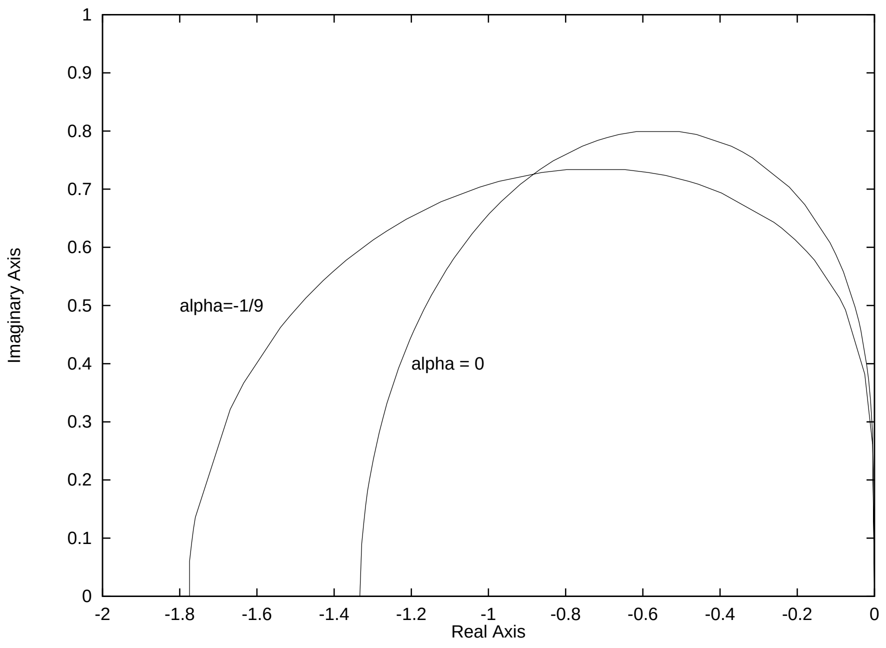


Figure 3.3: Explicit stability region.

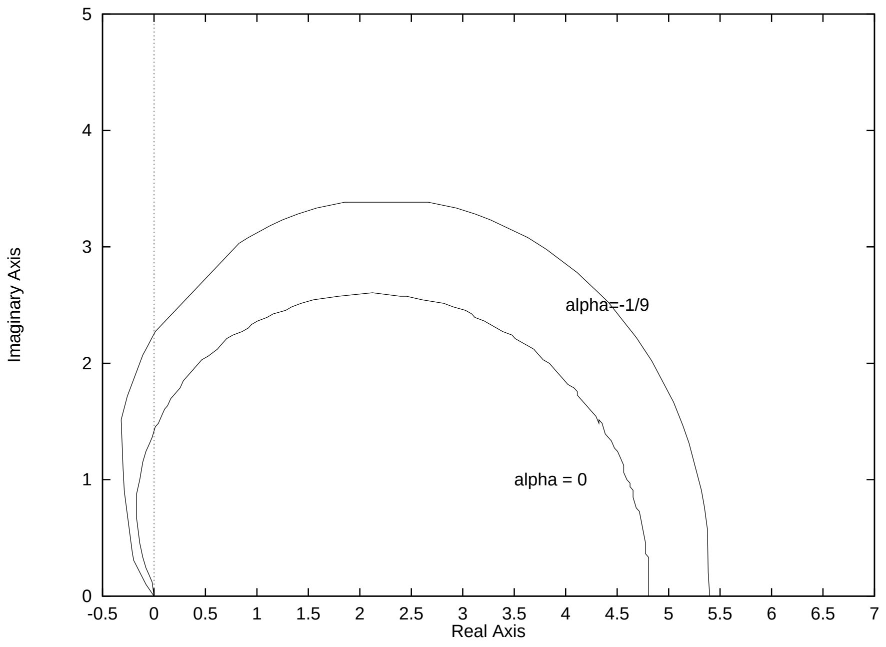


Figure 3.4: Modified stability region.

Figure (3.4) shows that NDF2 has a desirably smaller stability region than BDF2 in the right half plane(i.e. a larger instability region where the true solution is growing) but a less desirable slightly smaller one in the left half plane than BDF2 (see Section (3.5)). Both methods are A(α)-stable however. Now the convergence condition of the nonlinear equations splitting method as given in Section (3.6) for the model equation (3.49) used by [42] when both α and β are complex is given by

$$\mid [I \ - \ \triangle t\tilde{\gamma}\alpha]^{-1} \mid < \ \frac{r_c}{\tilde{\gamma}\triangle t \mid \beta \mid}, \tag{3.54}$$

see equation (3.48). The maximum values of $\triangle t \mid \beta \mid$ to be in the stability regions are 1.3 and 1.8 for BDF2 and NDF2, respectively, and $\tilde{\gamma} \ = \ \frac{2}{3}$ and $\tilde{\gamma}^* \ = \ 0.6$. Hence if $\triangle t\beta$ lies on the edge of the explicit region then $\tilde{\gamma}\triangle t \mid \beta \mid \approx \ 1$ and this requirement is very similar to the IMEX convergence requirements, which from equation (3.48) is given by

$$\mid [I \ - \ \triangle t\tilde{\gamma}\alpha]^{-1} \mid < \ r_c^{imex}. \tag{3.55}$$

Hence it is evident that in both cases $\alpha\triangle t$ must satisfy a very similar condition if the iterations are to be stable and to converge at the same rate.

3.9 Estimating the local splitting error

In Section (3.8) we have described two approaches to treat advection explicitly, when IMEX schemes are used. We have adopted the approach of Frank et al. [42] as given by equation (3.45). When this approach is being implemented, $\hat{\mathbf{V}}(t_{n+1})$ is the solution at the end of the step. As given by equation (3.26), $\mathbf{V}(t_{n+1})$ is an approximation to the solution at time t_{n+1}. Now we are interested in finding the local IMEX splitting error denoted by $\mathbf{e}\hat{\mathbf{V}}(t_{n+1})$ and defined by

$$\mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \ = \ [\mathbf{V}(t_{n+1}) \ - \ \hat{\mathbf{V}}(t_{n+1})]. \tag{3.56}$$

With the assumption that the past values $\mathbf{V}(t_j)$, $j \ = \ n, \ n-1, \ n-2$ for both methods are identical, and subtraction of equation(3.46) from equation(3.26) with the multiplication by $\tilde{\gamma}\triangle t$ and linearization gives

$$[I \ - \ \triangle t\tilde{\gamma}J_s]\mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \ = \ \tilde{\gamma}\triangle t[\mathbf{F}^f(t_{n+1}, \mathbf{V}(t_{n+1})) - \mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1}))]. \tag{3.57}$$

Then addition and subtraction of the term $\mathbf{F}^f(t_{n+1}, \ \hat{\mathbf{V}}(t_{n+1}))$ and further linearization gives that

$$\begin{aligned}[I \ - \ \triangle t\tilde{\gamma}J_s]\mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \ &= \ \tilde{\gamma}\triangle t[J_f(\mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \\ &+ \ \mathbf{F}^f(t_{n+1}, \ \hat{\mathbf{V}}(t_{n+1})) \ - \ \mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1}))],\end{aligned} \tag{3.58}$$

and in the above equation the higher order terms have been ignored. Multiplying with $[I \; - \; \triangle t\tilde{\gamma}J_s]^{-1}$ on both side of the above equation we have that

$$\begin{aligned} \| \, \mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \, \| \;\; &\leq \;\; \| \, [I \; - \; \triangle t\tilde{\gamma}J_s]^{-1}\tilde{\gamma}\triangle t[\mathbf{F}^f(t_{n+1}, V(t_{n+1})) - \mathbf{F}^f(t_{n-1}, \mathbf{V}^*(t_{n+1}))] \, \| \\ &+ \;\; r_c \, \| \, \mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \, \| \, . \end{aligned} \tag{3.59}$$

Then simplification gives that

$$\| \, \mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \, \| \leq \frac{\tilde{\gamma}\triangle t}{1 \; - \; r_c} \;\; \| \, [I - \triangle t\tilde{\gamma}J_s]^{-1}[\mathbf{F}^f(t_{n+1}, \mathbf{V}(t_{n+1})) - \mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1})] \, \| \, . \tag{3.60}$$

The righthand side term $\| \; . \; \|$ in the above equation may be calculated using one extra evaluation of $\mathbf{F}^f(t_{n+1}, \; \hat{\mathbf{V}}(t_{n+1}))$ and a backsolve. The equation (3.48) can be used to bound the norm of the inverse Jacobian $\| \; [I \; - \; \triangle t\tilde{\gamma}J_s]^{-1}$ by the rate of convergence r_c^{imex} to get

$$\| \, \mathbf{V}(t_{n+1}) - \hat{\mathbf{V}}(t_{n+1}) \, \| \leq r_c^{imex}\frac{\tilde{\gamma}\triangle t}{1 \; - \; r_c} \;\; \| \, [\mathbf{F}^f(t_{n+1}, V(t_{n+1})) - \mathbf{F}^f(t_{n+1}, V^*(t_{n+1}))] \, \| \, . \tag{3.61}$$

The right hand side term $\tilde{\gamma}\triangle \; \| \; . \; \|$ is related to the extra local truncation error due to splitting see [43], and can be written in the following form

$$\begin{aligned} -\triangle t[\mathbf{F}^f(t_{n+1}, \mathbf{V}(t_{n+1})) - \mathbf{F}^f(t_{n+1}, \mathbf{V}^*(t_{n+1}))] \;\; &= \;\; -\triangle t J_f(\hat{\mathbf{V}}(t_{n+1}) \; - \; \mathbf{V}^*(t_n)) \\ &+ \;\; h.o.t., \end{aligned} \tag{3.62}$$

where $\hat{\mathbf{V}}(t_{n+1}) \; - \; \mathbf{V}^*(t_{n+1}) \; = \; (\hat{\mathbf{V}}(t_{n+1}) \; - \; 2\mathbf{V}(t_n) \; + \; \mathbf{V}(t_{n-1}))$ may be regarded as an $O((\triangle t)^2)$ error. Alternatively with the help of equation (3.36) we can derive the following expression

$$\| \, \mathbf{e}\hat{\mathbf{V}}(t_{n+1}) \, \| \leq \frac{r_c}{1 \; - \; r_c} \, \| \, \hat{\mathbf{V}}(t_{n+1}) \; - \; \mathbf{V}^*(t_{n+1}) \, \| \, . \tag{3.63}$$

Then in both case the quantity r_c plays an vital role in the relationship between the IMEX splitting error and the quantity $\hat{\mathbf{V}}(t_{n+1}) \; - \; \mathbf{V}^*(t_{n+1})$. As the restriction on r_c is also required for convergence of the nonlinear splitting method, this restriction thus appears to be important to both methods for different reasons.

3.10 The Time Step Mechanism

The NDF and the theta code explained in this chapter used a variable step-size in which the time step is chosen automatically throughout the integration to satisfy a user specified local error tolerance. The NDF code uses the Newton iteration, while the theta method code implies either Newton iteration or functional iteration to solve the non-linear equations at each time step. For the selection of the step size the local error indicated by $\mathbf{le}(t_{n+1})$ is used and for the NDF code this error estimate is based on the difference between the predictor and corrector. Now consider the weighted error norm

$$\| \mathbf{le}(t_{n+1}) \|_w = \max \left(\frac{| le_j(t_{n+1}) |}{\overline{W}_j^n} \right), \tag{3.64}$$

where $\overline{W}_j^n = ATOL_j + RTOL_j V_j(t_n)$ and $ATOL_j$ and $RTOL_j$ the absolute and relative error tolerance for component j. An integration step is accepted if $\| \mathbf{le}(t_{n+1}) \|_w \leq 1$ and rejected otherwise. The following new step strategy has been adopted.

$$\begin{cases} rh = \frac{\triangle t}{\triangle t_{old}} \\ rh = max(rh, \frac{\triangle t_{min}}{|\triangle t|}) \\ rh = \frac{rh}{max(1.0,\ |\triangle t| \times \triangle t_{max} \times rh)} \\ \triangle t = rh * \triangle t \end{cases} \tag{3.65}$$

where $\triangle t_{min}$ and $\triangle t_{max}$ are bounds of the stepsize $\triangle t$, $\triangle t_{old}$ is the previous stepsize and rh is the growth factor. After the convergence of the nonlinear solver, the weighted local error norm is estimated and if this norm is < 1, then the solution satisfies the local error tolerance. Unfortunately if it is ≥ 1 then we must reduce the step size. In case the nonlinear equation solver fails to converge, then first we update the Jacobian matrix and even after that failure occurs then we think about the reduction of the step size if possible. For the theta method ([9]) we have used the doubling and halving the time step strategy.

3.11 Conclusion

We have applied the NDF2 method to solve the chemical kinetics arising from atmospheric chemistry. The analysis has shown that it has slightly superior accuracy and stability properties to the more widely used BDF2 method for the type of ODE systems considered here, and the step-size has increased by the factor of 1.26. We have used the adaptive Gauss-Seidel iterative method, instead of a Newton-type iterative method. The results have shown that with NDF2 method, the Gauss-Seidel iterations have reduced almost 30% as compared to Verwer [90]. In term of number of steps, at modest tolerance the NDF2 method takes a number of step comparable to that of Verwer [90] and at tighter tolerance, about 10% reduction of steps have occurred in all cases except one.

A stability analysis of IMEX schemes has been given for the NDF2 method called here the NDF2 IMEX scheme. The test equation used is that of Ascher et al.[4] as given by equation (3.49). Firstly we took the α and β in the same equation as reals, then extended the stability analysis by taking both α and β complex. In all cases it has been observed that NDF2 IMEX scheme has better stability region, as compared to BDF2 IMEX scheme. Besides this NDF2 IMEX scheme is stable on purely imaginary axis while BDF2 IMEX scheme not. A comparison between the IMEX and nonlinear splitting approaches has shown some interesting similarities and has indicated a way of measuring the IMEX splitting error. The advantage of the splitting approach- that it more closely couples the flow and chemistry-is balanced by its greater cost after the first two iterations if the local splitting error estimation is included.

The numerical experiments have shown that the new NDF2 code works well but have also indicated that some tuning of the stepsize strategy and Jacobian evaluation criteria may be needed. At present the approach described here is already being used successfully in large scale experiments in computational atmospheric modelling.

Chapter 4

The numerical Solution of a Model Problem

4.1 Introduction

A currently active area of research is the numerical approximation of PDEs with stiff non linear source terms [61, 81]. Tang [81] concentrated on the convergence analysis for operator-splitting methods when applied to conservation laws with stiff source terms. Papalexandris et al. [61] have considered the spatial discretization aspect when stiff source terms are involved. But here we have paid attention to the time aspect when there is a stiff source term involved in the PDE. Such problems come from the modelling the atmospheric chemistry, non-equilibrium gas dynamics and combustion. Earlier workers [19, 52] have shown that spurious numerical solution phenomenon may occur when the insufficient spatial and temporal resolution are used; both have reported that the incorrect wave speeds and incorrect discontinuities when the PDE has stiff source terms.

In this chapter we will focus on the Leveque and Yee problem [52] in both one and two space dimensional cases using the method of lines approach. Recall that in this method a suitable spatial discretisation scheme is applied to the advection term and the PDE is reduced to a system of ordinary differential equations (ODEs) in time. The factors that effect the performance of the method include the spatial discretisation error, the position of the spatial discretisation points and the time integration method.

In particular the spatial mesh points should be chosen to reflect the true solution of the PDE. After choosing the spatial mesh the next step is to pick an appropriate ODE solver. As mentioned in [7] there are many adaptive algorithms available for controlling the spatial discretisation error. Although these algorithms use the spatial error to refine and coarsen the mesh, the aim is to integrate in time with sufficient accuracy so that the spatial error is not degraded while maintaining the efficiency of the time integrator. This has been achieved by varying the time accuracy tolerance with spatial error rather than keeping it fixed (see [7, 49]). In the present work, the method of Berzins [7] developed for convection dominated PDEs in two space dimensions has been applied to the 1D and 2D PDE of Leveque and Yee [52]. The central idea is that the temporal error should not corrupt the spatial discretisation error.

When solving time-dependent PDEs the error introduced may be split into the temporal error and spatial error, [7]. The next task is to estimate and control them in a sophisticated way so that the method works efficiently. Two approaches for controlling local error in time are given in the literature (see [7, 12]). The first approach is related to controlling the local error in time per step while in the second approach the local error in time per unit step is controlled. A complete description can be found in [7, 12, 49]. As mentioned in Lawson et al. [49], controlling the local error in time per step does not reflect the true growth in the global error as well as controlling the local in time per unit step. This Chapter will thus explain how to implement the local error per unit step control strategy when a source term is present and to determine how the source term effects the spatial discretisation error. For the spatial discretisation error we will use a strategy similar to that of [12]. The local temporal error is controlled in such a way that it is the fraction of the spatial discretisation error over each step and the theta method [9] is used as the time integrator. This approach requires an estimate of the spatial error to be calculated and depends for its robustness on the quality of this estimate.

In order to deal with stiff source terms a Gauss Seidel iterative technique has been implemented to solve the systems of non-linear equations, because this iteration works well in a method of lines frame work, see [90]. The first part of the work deals with the implementation of these ideas for a 1D hyperbolic conservation law with a nonlinear source term, [52]. Extensive numerical experimentation has been done and the results are compared with the exact solution whenever possible. The second part deals with the implementation of aforementioned technique in 2D

Leveque and Yee problem [52].

4.2 Spatial Discretisation

In the present numerical investigation, the focus is on the 1D Leveque and Yee problem [52], which is given by

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = -\psi(u) \quad x \in [0, \infty), \tag{4.1}$$

where $\psi(u)$ is the source term defined by $\mu u(u-1)(u-0.5)$ and $f(u(x,t)) = u(x,t)$ and is the linear advection with a source term that is "stiff"for large μ. The characteristic decomposition of above equation gives, [61]

$$\frac{du}{dt} = -\psi(u) \quad \text{along} \quad \frac{dx}{dt} = \lambda, \tag{4.2}$$

where $\lambda = \frac{\partial f(u)}{\partial u}$ and is equal to 1.0. In the absence of the source term the equation (4.1) can be written as

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad x \in [0, \infty). \tag{4.3}$$

Now consider $x = x(t)$ as a function of t, then $u = u(x(t), t)$, because u is also a function of time, t, then the rate of change of u along x(t) is

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{dx}{dt}\frac{\partial u}{\partial x}. \tag{4.4}$$

Now the equation (4.2) gives $\frac{dx}{dt} = 1$, and from above equation we have that

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{dx}{dt}\frac{\partial u}{\partial x} = 0, \tag{4.5}$$

which shows that the rate of change of u along the characteristic curve $x(t)$ is zero, and which implies that u is constant along the curve $x(t)$.

The next question that arises is along which curve does u as defined by the equation (4.1) remains constantΓ The answer is

$$\frac{du}{dt} = 0 \quad \text{along } \frac{dx}{dt} = f_u - \frac{\psi(u)}{\frac{\partial u}{\partial x}}, \tag{4.6}$$

and details are given in [61]. The special feature of equation (4.1) is that the discontinuity will move with constant speed ($\lambda = 1$), and the shock will never be formed in this problem.

Although this model problem is inadequate as a full test of any numerical method, it reveals the essential difficulties that make it possible to understand the numerical problems, identify their source and yields insight that may be valuable in developing better numerical methods . To find the source of difficulty we compare the present situation with that of a homogeneous system of conservation laws with no source term. In the case of a homogeneous system of conservation laws with no source terms, the use of a conservative numerical methods guarantees the propagation of the discontinuity in the initial data at the correct speed. To see this, the cell average is defined (see [52]) as

$$U_j(t) \;=\; \frac{1}{\triangle x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t)dx, \tag{4.7}$$

where $U_j(t)$ is a numerical approximation to $u_j(t)$. The integration of the conservation law $u_t \;+\; f(u)_x \;=\; 0$, over the interval $[x_{j-\frac{1}{2}},\; x_{j+\frac{1}{2}}] \;\times\; [t_n,\; t_{n+1}]$ yields that

$$V_j(t_{n+1}) \;=\; V_j(t_n) \;-\; \frac{1}{\triangle x}\left[\int_{t_n}^{t_{n+1}} f(V(x_{j+\frac{1}{2}},t)dt \;-\; \int_{t_n}^{t_{n+1}} f(V_{j-\frac{1}{2}},t)dt\right], \tag{4.8}$$

where $V_j(t_n)$ is the approximation to $u(x_j,t)$ by the time integration method. Summation of the above expression over j (j is the number of the grids points) gives the cancellation of the flux term and we are left with only fluxes at the boundaries of our region. A finite difference is said to be conservative if it can be written in the conservation form

$$V_j(t_{n+1}) \;=\; V_j(t_n) \;-\; \frac{\triangle t}{\triangle x}\left[f_{j+\frac{1}{2}} \;-\; f_{j-\frac{1}{2}}\right]. \tag{4.9}$$

In equation (4.9) $f_{j\pm\frac{1}{2}}$ are the numerical fluxes based on V at neighbouring points and $\triangle t f_{j+\frac{1}{2}}$ approximates the corresponding integral in equation (4.8). The summation of equation (4.8) gives the same cancellation of flux as in the true solution. Now if we include the source term and integration of following equation

$$u_t \;+\; \frac{\partial f(u)}{\partial x} \;=\; \psi(u), \tag{4.10}$$

over the interval $[x_{j-\frac{1}{2}},\; x_{j+\frac{1}{2}}] \;\times\; [t_n,\; t_{n+1}]$ reveals that

$$\begin{aligned} V_j(t_{n+1}) \;&=\; V_j(t_n) \;-\; \frac{1}{\triangle x}\int_{t_n}^{t_{n+1}} f(V(x_{j+\frac{1}{2}},t))dt \;-\; \int_{t_n}^{t_{n+1}} f(V(x_{j-\frac{1}{2}},t)dt \\ &\quad+\; \frac{1}{\triangle x}\int_{t_n}^{t_{n+1}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \psi(V(x,t)dxdt. \end{aligned} \tag{4.11}$$

It is evident from the equation that the new term appearing in equation (4.11) does not undergo cancellation while summing over j. This is main source of difficulty while solving problems with stiff source term, consequently it is important that this term is modelled accurately if we are to obtain the correct behaviour (see for example [52]).

4.3 Space-Time Error Balancing Control.

While solving PDEs a common approach is to select the time step in such a way that the CFL condition is satisfied. This guarantees the stability of the methods, but on the other hand the solution may not be of required accuracy, [62]. Hence the option of using the approach of controlling the local growth in time of the spatial discretization error is preferred [62]. It is mentioned in [62] that when the local time error has been controlled to required accuracy, the CFL condition will automatically be satisfied in some sense. The disadvantage of this type of error control strategy is that there is no clear relationship between the accuracy tolerance and the global space and time error [7, 12].

Our aim here is to develop an error control strategy that allows the accuracy tolerance to be selected and adjusted automatically for problems involving chemistry source terms. This is difficult with the local error per step control strategy because the time global error is not proportional to the local error tolerance (tol) see for example [7, 49]). We are interested in solving the initial value problem given by

$$\dot{\mathbf{U}} = \mathbf{F}_N(t, \mathbf{U}(t)), \qquad \mathbf{U}(0) \text{ given}, \tag{4.12}$$

nd its true solution $[U(t_n)]_{n=0}^{\tilde{p}}$ is approximated by $[V(t_n)]_{n=0}^{\tilde{p}}$ at set of discrete time points $0 = t_0 < t_1 < ... < t_{\tilde{p}} = t_f$ by the time integrator. The vector values of the global error at the spatial mesh points, at any time t is denoted by $\hat{\mathbf{E}}(t)$ and defined by (see for example [49])

$$\hat{\mathbf{E}}(t) = \mathbf{u}(t) - \mathbf{V}(t), \tag{4.13}$$

where $\mathbf{u}(t)$ represents the the restriction of the exact PDE solution to mesh, i.e.,

$$[\mathbf{u}(t)]_j = u(x_j, t), \qquad j = 1, ..., N. \tag{4.14}$$

The vector $\hat{\mathbf{E}}(t)$ can be written as the combination of the restriction of the PDEs spatial discretization error $\mathbf{est}(t)$ and the ODE global error $\mathbf{ge}(t, tol)$

$$\hat{\mathbf{E}}(t) = \mathbf{est}(t) + \mathbf{ge}(t, tol), \tag{4.15}$$

where $\mathbf{est}(t)$, the restriction of the PDEs spatial discretisation error, is given by

$$\mathbf{est}(t) = \mathbf{u}(t) - \mathbf{U}(t), \tag{4.16}$$

and the ODEs global error can be written as

$$\mathbf{ge}(t_{n+1}, tol) = \tilde{\mathbf{W}}(t_{n+1}) + \mathbf{le}(t_{n+1}), \tag{4.17}$$

see [7, 49] for details. In the above equation $\tilde{\mathbf{W}}(t_{n+1})$ represents the solution of the variational equation given by (see Shampine [73])

$$\dot{\tilde{\mathbf{W}}} = J\tilde{\mathbf{W}}, \qquad \tilde{\mathbf{W}}(t_n) = \mathbf{ge}(t_n, tol), \tag{4.18}$$

where $J = \frac{\partial F_N(\mathbf{U})}{\partial \mathbf{U}}$ and time local error $\mathbf{le}(t_{n+1})$ has been explained in Chapter 3 and also see [7, 9, 63] for more detail. Combining equations (4.15) and (4.17) the global time error can be written as,(see [7, 49])

$$\hat{\mathbf{E}}(t_{n+1}) = \mathbf{est}(t_{n+1}) \; + \; \tilde{\mathbf{W}}(t_{n+1}) + \mathbf{le}(t_{n+1}). \tag{4.19}$$

Since we are interested in both global and local accuracy and equation (4.19) shows that the true relation between the time global error and the user supplied tolerance is not clear. Hence controlling the local error does not guarantee the equal control of the global error. When solving PDEs it is important that the error control strategy must ensure that the time global error must be proportional to the the required accuracy. This is said to be case (see Berzins [7]) if global error at time t for an accuracy requirement tol is $\hat{E}_{tol}$, then for $\tilde{r} > 0$,

$$\hat{E}_{tol\times\tilde{r}} \;=\; \tilde{r} \times \hat{E}(t)_{tol}. \tag{4.20}$$

It is also shown in [7, 49] that this proportionality can be achieved if and only if the local error $\mathbf{le}(t_{n+1})$ for the given tolerance(tol) satisfies

$$\mathbf{le}(t_{n+1}) = \triangle t\overline{\gamma}(t_{n+1}, t_n)tol + O(\triangle t, tol), \tag{4.21}$$

where the behaviour of $\overline{\gamma}$ is very similar to integral mean over [τ, t] of a function that is independent of tol and bounded on [0, t_f]. Here the term $O(\triangle t, tol)$ is numerically negligible compared with terms of order $\triangle t$tol in the above equation.

So the technique used in [7, 49] in which the user supplied tolerance is related to the spatial discretisation error in some way is given as

$$\| \mathbf{le}(t_{n+1}) \| \; = \; \epsilon \, \| \hat{\mathbf{est}}(t_{n+1}) \| \, . \tag{4.22}$$

In the above equation ϵ is balancing factor and $\hat{\mathbf{est}}(t_{n+1})$ represents the local growth in time of the spatial discretisation error from t_n to t_{n+1}. The local growth in time of the spatial discretization error is defined as the spatial error at time t_{n+1} given the assumption that the spatial error $\mathbf{est}(t_n)$ at time t_n is zero (see for example [12]). The next task is to estimate the local growth in time of the spatial discretization error. For this purpose the procedure developed in (Berzins and Ware [12]) has been extended to PDEs with stiff source terms.

The underlying idea is to evaluate the primary solution using one upwind scheme applied to the advection term and a quadrature rule to integrate the stiff source term. Then the secondary solution is being estimated at same time step with upwind scheme of different order and different quadrature rule for source term integration. The difference between the two computed solutions has been taken as an estimate of the local growth in time of the spatial discretization error in the same way as in [7, 49]. Suppose that the ODE function in equation (4.12) at time t_n is given by

$$\mathbf{F}_N(t_n, \mathbf{U}(t_n)) \; = \; \mathbf{F}_N^f(t_n, \mathbf{U}(t_n)) + \mathbf{F}_N^s(t_n, \mathbf{U}(t_n)), \tag{4.23}$$

where $\mathbf{F}_N^f(t_n, \mathbf{U}(t_n))$ is the discretisation of the convective flux terms in equation (4.1) and with the implementatation of the second order upwind together with a suitable limiter at jth grid point is denoted $\tilde{F}_j^f(t_n, U_j(t_n))$ and given by

$$\tilde{F}_j^f(t_n, U_j(t_n)) \; = \; -\frac{1}{\triangle x}\left[1 + \frac{1}{2}\left(B(r_j, 1) - \frac{1}{r_{j-1}} B(r_{j-1}, 1)\right)\right](U_j(t_n) - U_{j-1}(t_n)), \tag{4.24}$$

where B(.,.) is any suitable limiter (see Chapter 2 and [7, 49]) and the factor r_j is defined by (see equation (2.65))

$$r_j \; = \; \frac{U_{j+1}(t_n) - U_j(t_n)}{U_j(t_n) - U_{j-1}(t_n)}. \tag{4.25}$$

$\mathbf{F}_N^s(t_n, \mathbf{U}(t_n))$ represents the source term integration in the same equation and at jth grid point we have that

$$\begin{aligned} F_j^s(t_n, U_j(t_n)) \; &\approx \; -\frac{1}{\triangle x}\phi(U_j(t_n)), \\ -\frac{1}{\triangle x}\phi(U_j(t_n)) \; &= \; -\frac{1}{\triangle x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \psi(U_j(t_n))dx, \end{aligned} \tag{4.26}$$

where $\phi(U_j(t_n))$ represents the integration of source term at jth grid point with midpoint rule and is given as

$$\phi(U_j(t_n)) = \triangle x \psi(U_j(t_n)). \tag{4.27}$$

Then at jth grid point the $F_j^s(t_n, U_j(t_n))$ is given by

$$F_j^s(t_n, U_j(t_n)) \;=\; -\psi(U_j(t_n)). \tag{4.28}$$

So the equation (4.23) with the help of equations (4.24) and (4.28) at jth grid point can be written as

$$\begin{aligned} F_j(t_n, U_j(t_n)) \;&=\; -\frac{1}{\triangle x}\left[1 + \frac{1}{2}\left(B(r_j, 1) - \frac{1}{r_{j-1}} B(r_{j-1}, 1)\right)\right](U_j(t_n) - U_{j-1}(t_n)) \\ &-\; \psi(U_j(t_n)). \end{aligned} \tag{4.29}$$

Now the auxiliary solution used for the evaluation of the local growth in time of the spatial discretization error at time t_n is calculated by the solution of following modified ODEs system

$$\dot{\mathbf{v}}_{n+1}(t_n) \;=\; \mathbf{G}_N(t_n, \mathbf{v}_{n+1}(t_n)), \tag{4.30}$$

where $\mathbf{v}_{n+1}(t_n) = \mathbf{V}(t_n)$. In the above equation $\mathbf{G}(t_n, \mathbf{V}(t_n))$ has the following form

$$\mathbf{G}_N(t_n, \mathbf{V}(t_n)) \;=\; \mathbf{G}_N^f(t_n, \mathbf{V}(t_n)) + \mathbf{G}_N^s(t_n, \mathbf{V}(t_n)), \tag{4.31}$$

where $\mathbf{G}_N^f(t_n, \mathbf{V}(t_{n+1}))$ is the spatial discretization with the upwind scheme of order different from equation (4.24) and when the limiter function B(.,.) is zero (see equations(2.56) and (2.57)), means first order upwind scheme, has the following form at the jth grid point

$$G_j^f(t_n, V_j(t_n)) \;=\; -\frac{V_j(t_n) - V_{j-1}(t_n)}{\triangle x}. \tag{4.32}$$

Now $\mathbf{G}_N^s(t_n, V_j(t_n))$ is evaluated by the trapezoidal rule and at jth grid given as

$$\begin{aligned} G_j^s(t_n, V_j(t_n)) \;&\approx\; -\frac{1}{\triangle x}\tilde{\phi}(V_j(t_n)), \\ -\frac{1}{\triangle x}\tilde{\phi}(V_j(t_n)) \;&=\; -\frac{1}{\triangle x}\int_{x-\frac{1}{2}}^{x+\frac{1}{2}} \psi(V(t_n))dx. \end{aligned} \tag{4.33}$$

In the above equation

$$\tilde{\phi}(V_j(t_n)) \;=\; \frac{\triangle x}{2}\left(\psi(V_{j-\frac{1}{2}}(t_n)) \;+\; \psi(V_{j+\frac{1}{2}}(t_n))\right), \tag{4.34}$$

where

$$\begin{aligned}\psi(V_{j-\frac{1}{2}}(t_n)) &= \frac{\psi(V_j(t_n)) + \psi(V_{j-1}(t_n))}{2},\\ \psi(V_{j+\frac{1}{2}}(t_n)) &= \frac{\psi(V_j(t_n)) + \psi(V_{j+1}(t_n))}{2}.\end{aligned}$$

So with help of above equation the equation (4.33) can be written as

$$G_j^s(t_n, V_j(t_n)) = -\frac{1}{4}\left(\psi(V_{j-1}(t_n)) + 2\psi(V_j(t_n)) + \psi(V_{j+1}(t_n))\right). \tag{4.35}$$

So with the combination of equations (4.30) and (4.35) the function $\mathbf{G}(t_n, \mathbf{V}(t_n))$ (see equation (4.31)) at jth grid point is given by

$$\begin{aligned}G_j(t_n, V_j(t_n)) &= -\frac{V_j(t_n) - V_{j-1}(t_n)}{\triangle x}\\ &- \frac{1}{4}\left(\psi(V_{j-1}(t_n)) + 2\psi(V_j(t_n)) + \psi(V_{j+1}(t_n))\right).\end{aligned} \tag{4.36}$$

Then the local in time of the spatial discretization error is estimated by

$$\hat{\mathbf{est}}(t_{n+1}) = \mathbf{V}(t_{n+1}) - \mathbf{v_{n+1}}(t_{n+1}). \tag{4.37}$$

Since only the order of the magnitude of the norm of the local growth in time of the spatial discretization error is needed, it is sufficient to apply only a few Gauss Seidel iterations to compute the auxiliary solution of the ODE system given by (4.30). If the theta method (Chapter 3) is being used as the time integrator then this equation in combination with equation(4.37) and equation (3.7) gives

$$\begin{aligned}\hat{\mathbf{est}}(t_{n+1}) &= \triangle t\theta\left(\mathbf{F}_N(t_{n+1}, \mathbf{V}(t_{n+1})) - \mathbf{G}_N(t_{n+1}, \mathbf{V}(t_{n+1}))\right) \qquad (4.38)\\ &+ \triangle t(1-\theta)\left(\mathbf{F}_N(t_n, \mathbf{V}(t_n)) - \mathbf{G}_N(t_n, \mathbf{V}(t_n))\right).\end{aligned}$$

When the solution and the time derivative calculated by equation (4.12) is substituted in equation (4.30), the residual of the auxiliary equation can be written as

$$\mathbf{r}(t_n, \mathbf{V}(t_n)) = \dot{\mathbf{V}}(t_n) - \mathbf{G}(t_n, \mathbf{V}(t_n)). \tag{4.39}$$

Using the equations (4.12) and (4.39) and substituting for $\mathbf{F}_N(t_n, \mathbf{V}(t_n))$ using $\dot{\mathbf{V}}(t_n)$ gives

$$\hat{\mathbf{est}}(t_{n+1}) = \triangle t\left(\theta\mathbf{r}(t_{n+1}, \mathbf{V}(t_{n+1})) + (1-\theta)\mathbf{r}(t_n, \mathbf{V}(t_n))\right). \tag{4.40}$$

So the time tolerance when the strategy of the LEPUS is being implemented on the step t_{n+1} is given by

$$TOL = \epsilon \parallel \theta\mathbf{r}(t_{n+1}, \mathbf{V}(t_{n+1})) + (1-\theta)\mathbf{r}(t_n, \mathbf{V}(t_n)) \parallel. \tag{4.41}$$

Now $\mathbf{F}_N(t_n, \mathbf{V}(t_n))$ is approximated by the second order upwind scheme, while the first order upwind scheme is used for $\mathbf{G}_N(t_n, \mathbf{V}(t_n))$, so a simple calculation shows that equation (4.39) can be written as

$$\begin{aligned} r_j(t_n, V(t_n)) &= -\frac{\triangle V_{j-\frac{1}{2}}(t_n)}{2\triangle x}\left[B(\frac{1}{r_j}, 1) - \frac{1}{r_{j-1}}B(r_{j-1}, 1)\right] \\ &+ \frac{1}{4}\left(\psi(V_{j+1}(t_n)) - 2\psi(V_j(t_n)) + \psi(V_{j-1}(t_n))\right), \qquad (4.42) \end{aligned}$$

where $\triangle V_{j-\frac{1}{2}}(t_n) = V_j(t_n) - V_{j-1}(t_n)$ and $r_j(t_{n+1}, V(t_{n+1}))$ can be expressed by an almost identical form.

4.3.1 Local Growth in Spatial Discretization Error

As explained in the previous Section the approach used is to calculate the primary solution with one particular upwind scheme applied to the advection term and one quadrature rule used to integrate the stiff source term. The secondary solution is then obtained with upwind scheme of different order and a different quadrature rule. The difference between the two solutions is an indicator of the local growth in time of the spatial discretization error (see [7]). Hence with the implementation of the second order upwind scheme to advection term and midpoint rule to the source term the ODE function as given by equation (4.23) at jth grid point can be written as

$$\begin{aligned} F_j(t_n, U(t_n)) &= -\frac{1}{\triangle x}\left[1 + \frac{1}{2}\left(B(r_j, 1) - \frac{1}{r_{j-1}}B(r_{j-1}, 1)\right)\right](U_j(t_n) - U_{j-1}(t_n)) \\ &- F_j^s(t_n, U_j(t_n)), \qquad (4.43) \end{aligned}$$

where $F_j^s(t_n, U_j(t_n))$ is the source term integration with midpoint rule, (see equation (4.28)). For the time being we have taken the factor r defined in equation (4.25) to be constant then $B(r, 1) - \frac{1}{r}B(r, 1) = \frac{(r+|r|)(r-1)}{r(1+r)}$ where B(.,.) is the van Leer limiter given as (see [7])

$$B(r, 1) = \frac{r + \mid r \mid}{1 + r}, \qquad (4.44)$$

and with the help of Taylor series $U_j(t_n) - U_{j-1}(t_n) = \triangle x\frac{\partial U_j(t_n)}{\partial x}$. The ODEs function as given by equation(4.43) has the following form

$$\begin{aligned} F_j(t, U_j(t)) &= -\frac{1}{\triangle x}\left[1 + \frac{1}{2}\frac{(r+\mid r \mid)(r-1)}{r(1+r)}\right](U_j(t_n) - U_{j-1}(t_n)) \\ &- F_j^s(t_n, U_j(t_n)). \qquad (4.45) \end{aligned}$$

For simplicity we define

$$\hat{a} \;=\; \left[\frac{(r+\mid r\mid)(r-1)}{r(1+r)}\right], \tag{4.46}$$

which is zero for r negative, and when r is positive has the value $\hat{a} \;=\; \left[2\frac{(r-1)}{(1+r)}\right]$, and next task is to simplify $\hat{a}$ by making use of the definition of r which is given by equation (4.25)

$$r \;=\; \frac{U_{j+1}(t_n)-U_j(t_n)}{U_j(t_n)-U_{j-1}(t_n)}, \tag{4.47}$$

$$r+1 \;=\; \frac{U_{j+1}(t_n)-U_{j-1}(t_n)}{U_j(t_n)-U_{j-1}(t_n)},$$

and

$$r-1 \;=\; \frac{U_{j+1}(t_n)-2U_j(t_n)+U_{j-1}(t_n)}{U_j(t_n)-U_{j-1}(t_n)}.$$

From this it can be inferred that $\hat{a}$ in equation (4.46) has the following form

$$\hat{a} \;\approx\; 2\frac{(\triangle x)^2(U_j)_{xx}(t_n)}{2\triangle x(U_j)_x(t_n)}, \tag{4.48}$$

and further simplification gives that

$$\hat{a} \;\approx\; \frac{(\triangle x)(U_j)_{xx}(t_n)}{(U_j)_x(t_n)}. \tag{4.49}$$

With this value of $\hat{a}$, the truncation error associated with the equation (4.45) may be written as, just for convience, we denote it with SDE_1

$$SDE_1 \;=\; -\left[(1+\frac{(\triangle x)^2(U_j)_{xx}(t_n)}{2\triangle x TE_x}\right]TE_x - O(\triangle x^2), \tag{4.50}$$

where $O(\triangle x)^2$ is the error due to mid point rule, which is applied for the source term integration and TE_x truncation error being introduced due to the spatial discretization of $\frac{\partial U_j(t_n)}{\partial x}$ and will be of the form $\triangle x\frac{\partial^2 U_j(t_n)}{\partial x^2}$.

The next task is to find the auxiliary solution with upwind scheme of different order and different quadrature rule for source term integration that have have applied to find the primary solution. So with the implementation of first order upwind scheme and the trapezoidal rule (see equation (4.34)) to the source term integration the auxiliary ODE function at jth grid point is given by

$$G_j(t_n, U_j(t_n)) \;=\; -\frac{U_j(t_n)-U_{j-1}(t_n)}{\triangle x} - G_j^s(t_n, U_j(t_n)), \tag{4.51}$$

where $G_j^s(t_n, U_j(t_n))$ is the integration of the source term with the trapezoidal rule (see equation (4.35)), then the truncation error will be of the form

$$SDE_2 = -TE_x + O(\triangle x^2), \tag{4.52}$$

where $O(\triangle x)^2$ is error due to the trapezoidal rule, as used for the source term integration. The error TE_x being introduced due to the spatial discretization will be of the form $\triangle x \frac{\partial^2 U_j(t_n)}{\partial x^2}$ we have conviently denoted it by SDE_2.

If the Backward Euler Method is being implemented as the time integrator then the difference between equation (4.50) and equation (4.52) is the local growth in time of the spatial discretization error and is given below at jth grid point

$$\hat{est}_j(t_{n+1}) = -\triangle t \left[\frac{\triangle x}{2} \frac{\partial^2 V_j(t_{n+1})}{\partial x^2} \right] - \triangle t \; O(\triangle x)^2), \tag{4.53}$$

where $V_j(t)$ is the numerical approximation to $U_j(t)$ with the time integrator and the second term is due to the source. It is evident that local growth in time of the spatial discretization error will increase with an increasing in stiffness of the source term. The reason for this is that the second error term in the above equation appearing due to the source term, which will increase in size with increasing stiffness.

4.4 Stability for the Scheme

As discussed earlier, using the upwind scheme with a zero limiter for the spatial discretisation, we have the following form of equation (4.1) at jth grid point and time t

$$\frac{\partial U_j}{\partial t} = -\frac{(U_j(t) - U_{j-1}(t))}{\triangle x} - F_j^s(t, U_j(t)), \tag{4.54}$$

and $F^s(t_n, U_j(t_n))$ represents the integration of the source term with mid point rule (see equation (4.28)). When the limiter is not zero, we have the following equation

$$\begin{aligned} \frac{\partial U_j}{\partial t} &= -\frac{1}{\triangle x}\left[1 + \frac{1}{2}\left(B(r_j, 1) - \frac{1}{r_{i-1}} B(r_{j-1}, 1)\right)\right](U_j(t) - U_{j-1}(t)) \\ &- F^s(t, U_j(t)). \end{aligned} \tag{4.55}$$

Using the vector notation the equation(4.55) takes the following form (see [7])

$$\dot{\mathbf{U}}(t) = -\frac{\left(I + \overline{B}(t, \mathbf{U}(t))\right)\overline{A}_N}{\triangle x}\mathbf{U}(t) - \psi(\mathbf{U}(t)), \tag{4.56}$$

and in the above equation $\overline{A}_N$ represents the first order upwind approximation and $\overline{B}$ stands for the term involving the limiter function. Let us define matrix C_N as a product of the matrices $\overline{A}_N$ and $(I+\overline{B})$ (diagonal matrix) and substituting in equation (4.56) gives

$$\dot{\mathbf{U}}(t) = -\frac{1}{\triangle x} C_N \mathbf{U}(t) - \mu\tilde{\psi}(\mathbf{U}(t)), \tag{4.57}$$

where $\tilde{\psi}$ has defined in equation (4.68). Let vector $\mathbf{V}(t_n)$ be the numerical approximation to $\mathbf{U}(t_n)$ generated by the time integrator, we have

$$\mathbf{V}(t_{n+1}) = \mathbf{V}(t_n) - \frac{\triangle t}{\triangle x} C_N \mathbf{V}(t_n) - \mu\triangle t\tilde{\psi}(\mathbf{V}(t_{n+1})). \tag{4.58}$$

In the above equation the source term is to be treated implicitly while an explicit technique has been used for the advection term. If $\mathbf{V}(t_n)$ is the numerical approximation to $\mathbf{U}(t_n)$ given by the time integrator, we define the growth of error in time $\tilde{\mathbf{E}}^n = \mathbf{U}(t_n) - \mathbf{V}(t_n)$ and applying the Mean Value Theorem to the source it gives

$$\tilde{\mathbf{E}}^{n+1} = -\frac{\triangle t}{\triangle x} C_N \tilde{\mathbf{E}}^n - \mu\triangle t\tilde{\mathbf{E}}^{n+1} J_s, \tag{4.59}$$

where J_s is the Jacobian matrix and defined by $\left[\frac{\partial\tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial\mathbf{V}(t_{n+1})}\right]$. Rearranging equation (4.59) we have

$$\tilde{\mathbf{E}}^{n+1} = -\frac{\triangle t}{\triangle x}(I + \mu\triangle t J_s)^{-1} C_N \tilde{\mathbf{E}}^n. \tag{4.60}$$

Hence in the presence of the stiff source term the stability requirement is

$$\left\| \left[\frac{\triangle t}{\triangle x}(I + \mu\triangle t J_s)^{-1} C_N\right] \right\| < 1, \tag{4.61}$$

thus giving a CFL type condition. For the non-stiff case the term $\triangle t\mu J_s$ can be ignored because it will be small, so we have that

$$\frac{\triangle t}{\triangle x} \| C_N \| < 1, \tag{4.62}$$

which is the similar stability condition as given by Berzins [7]. As the matrix $\frac{1}{\triangle x}C_N$ corresponds to spatial discretization of the advection term, the stability condition reduces to the standard CFL stability condition

$$\frac{\triangle t}{\triangle x} < 1. \tag{4.63}$$

While this CFL, condition provides a stable time step, there is no guarantee of accuracy. In the stiff source term case when the term $\triangle t\mu J_s$ is large the source term will act as a relaxation factor and allow a larger value of $\triangle t$ to be used.

4.5 Wave Speed

As mentioned earlier, our focus is on the Leveque and Yee [52] problem given by equation (4.1). In this Section the phenomenon of the numerical propagation of the front will be explored. Leveque and Yee solved this problem with the split algorithm and have seen no spurious wave speeds for $\lambda = 0.5$. From numerical experiments it is evident that the performance of the algorithm depends upon the critical dimensionless parameters; the mesh ratio $c = \triangle t/\triangle x$ and product $\triangle t\mu$ of the time step and reaction rate. Now we will calculate the step size using new approach.

The ODEs function as given by equation (4.12) can be written as

$$F_j(t, U_j(t)) \;=\; -\frac{(U_j(t) - U_{j-1}(t))}{\triangle x} - F_j^s(U_j(t)), \quad j \;=\; 1 \cdots NPTS \tag{4.64}$$

for simplicity the first order upwind method has been used for spatial discretization and the source term has been integrated with mid point rule over the control volume, so

$$F_j^s(U_j(t)) \;=\; \frac{1}{\triangle x}\phi(U_j(t)), \tag{4.65}$$

where $\phi(.)$ has been defined by equation (4.27). Then the integration of the source term is given by

$$F_j^s(U_j(t)) \;=\; \psi(U_j(t)). \tag{4.66}$$

For this analysis we have taken source term of the form

$$\psi(\mathbf{U}) = \mu\tilde{\psi}, \tag{4.67}$$

where

$$\tilde{\psi}(\mathbf{U}) = \mathbf{U}(1 \;-\; \mathbf{U}), \tag{4.68}$$

has been defined for notational convience below. The derivative of source term with respect to $\mathbf{U}$ is given by

$$\frac{\partial\psi(\mathbf{U})}{\partial\mathbf{U}} \;=\; \mu(1 \;-\; 2\mathbf{U}). \tag{4.69}$$

To reduce any unnecessary complexity in the analysis we have utilised the Backward Euler method as the time integrator and the Forward Euler as the predictor, getting

$$V_j^p(t_{n+1}) \;=\; V_j(t_n) + \triangle tF(t_n, V_j(t_n)), \tag{4.70}$$

and

$$V_j^c(t_{n+1}) \;=\; V_j(t_n) + \triangle tF(t_{n+1}, V_j(t_{n+1})). \tag{4.71}$$

In order to explain how the error balancing algorithm defined in equation (4.22) works when applied to the Leveque and Yee [52] we now need to obtain the explicit form of the time local error and the local in time space error.

4.5.1 Estimation of Time Local Error

The local error at t_{n+1} is defined as $\mathbf{le}(t_{n+1})$, with the jth component given by $le_j(t_{n+1})$, and is estimated in standard ODE codes by [63]

$$le_j(t_{n+1}) \;=\; \frac{V_j^p(t_{n+1}) - V_j^c(t_{n+1})}{2}. \tag{4.72}$$

Using equations (4.70) and (4.71), the above equation may be written as

$$\begin{aligned} le_j(t_{n+1}) \;&=\; \triangle t \frac{[(V_j(t_{n+1}) - V_{j-1}(t_{n+1})) - (V_j(t_n) - V_{j-1}(t_n))]}{2\triangle x} \\ &+\; \frac{\triangle t}{2}\left[\psi(V_j(t_{n+1})) - \psi(V_j(t_n))\right]. \end{aligned} \tag{4.73}$$

Rearranging gives

$$\begin{aligned} le_j(t_{n+1}) \;&=\; \triangle t \frac{[(V_j(t_{n+1}) - V_j(t_n)) - (V_{j-1}(t_{n+1}) - V_{j-1}(t_n))]}{2\triangle x} \\ &+\; \frac{\triangle t}{2}\left[\psi(V_j(t_{n+1})) - \psi(V_j(t_n))\right]. \end{aligned} \tag{4.74}$$

From the Mean Value Theorem we have

$$V_j(t_{n+1}) - V_j(t_n) \;=\; \triangle t \frac{\partial V_j(t_{n+1})}{\partial t} + O(\triangle t)^2. \tag{4.75}$$

Similarly we also have

$$V_{j-1}(t_{n+1}) - V_{j-1}(t_n) \;=\; \triangle t \frac{\partial V_{j-1}(t_{n+1})}{\partial t} + O(\triangle t)^2. \tag{4.76}$$

So the equation (4.74) takes the following form

$$\begin{aligned} le_j(t_{n+1}) \;&=\; \frac{\triangle t^2}{2} \frac{\left[\frac{\partial V_j(t_{n+1})}{\partial t} - \frac{\partial V_{j-1}(t_{n+1})}{\partial t}\right]}{\triangle x} \\ &+\; \frac{\triangle t}{2}\left[\psi(V_j(t_{n+1})) - \psi(V_j(t_n))\right]. \end{aligned} \tag{4.77}$$

Applying of the Taylor series expansion to the first term and the Mean Value Theorem to the second term gives

$$le_j(t_{n+1}) = \frac{\triangle t^2}{2}\left[\frac{\partial^2 V_j(t_{n+1})}{\partial t \partial x} + \frac{\partial V_j}{\partial t}\frac{\partial \psi(V_j(t_{n+1}))}{\partial V_j}\right] \;+\; O(\triangle t)^3, \tag{4.78}$$

and by substituting for $\frac{\partial\psi(V_j(t_{n+1}))}{\partial V_j(t_n)}$ from the equation (4.69) the equation (4.78) has the following form

$$le_j(t_{n+1}) = \frac{\triangle t^2}{2}\left[\frac{\partial^2 V_j(t_{n+1})}{\partial t\partial x} + \mu(1-2V_j(t_{n+1}))\frac{\partial V_j(t_{n+1})}{\partial t}\right] + O(\triangle t)^3, \quad (4.79)$$

and with help of equation (4.57) the above equation can be written as

$$\begin{aligned}\mathbf{le}(t_{n+1}) &= \frac{\triangle t^2}{2}[\frac{C_N(\frac{C_N\mathbf{V}(t_{n+1})}{\triangle x} + \mu\tilde{\psi}(\mathbf{V}(t_{n+1}))}{\triangle x} \qquad (4.80)\\ &+ \mu\tilde{\psi}_V\left(\frac{C_N\mathbf{V}(t_{n+1})}{\triangle x} + \mu\tilde{\psi}(\mathbf{V}(t_{n+1}))\right)] + O(\triangle t)^3,\end{aligned}$$

where $\tilde{\psi}$ has been defined in equation (4.68).

4.5.2 Estimation of Local in Time Space Error

The next task is to estimate the local growth in time of spatial discretization error. As explained earlier, the basic technique is to evaluate the primary solution by using one of upwind scheme for the advection term and a quadrature rule to integrate the stiff source term. Then the secondary solution is being estimated at same time step with an upwind scheme of different order and a different quadrature rule for source term integration. The difference between the two computed solutions is then used as an estimate of the local growth in time of the spatial discretization error.

Hence, for evaluating the primary solution, the limited second order upwind method has been used as the spatial discretization and the trapezoidal rule (equation (4.34)) is used to evaluate the source term. Then the ODEs function, (see equation (4.12)) is given by

$$\begin{aligned}F_j(t,U_j(t)) &= -\frac{1}{\triangle x}\left[1+\frac{1}{2}\left(B(r_j,1) - \frac{1}{r_{j-1}}B(r_{j-1},1)\right)\right](U_j(t) - U_{j-1}(t))\\ &- \tilde{F}_j^s(t,U_j(t)), \qquad (4.81)\end{aligned}$$

where $\tilde{F}_j^s(t,U_j(t))$ represents the source term integration with trapezoidal rule and is given by $\frac{1}{4}(\psi(U_{j-1}(t)) + 2\psi(U_j(t)) + \psi(U_{j+1}(t)))$ (see equation (4.35)). With the implementation of the Backward Euler as the time integrator, the difference of the equations (4.64) and (4.81) is taken as the estimate of the local growth in time of the spatial discretization error and is given at the jth grid point by

$$\begin{aligned}\hat{est}_j(t_{n+1}) &= \frac{\triangle t}{2\triangle x}\left[B(r_j,1) - \frac{1}{r_{j-1}}B(r_{j-1},1)\right](V_j(t_{n+1}) - V_{j-1}(t_{n+1}))\\ &+ \frac{\triangle t}{4}(\psi(V_{j-1}(t_{n+1})) - 2\psi(V_j(t_{n+1})) + \psi(V_{j+1}(t_{n+1}))). \qquad (4.82)\end{aligned}$$

Now given that

$$\begin{aligned} B(r,1) - \tfrac{1}{r}B(r,1) &= \tfrac{(r+|r|)(r-1)}{r(1+r)}, \\ V_j(t_n) \ - \ V_{j-1}(t_n) &\approx \triangle x \tfrac{\partial V_j(t_n)}{\partial x}, \end{aligned}$$

and assuming

$$\psi(V_{j-1}(t_{n+1})) \ - \ 2\psi(V_j(t_{n+1})) \ + \ \psi(V_{j+1}(t_{n+1})) \ = \ \triangle x^2 \frac{\partial^2(\psi(V_j(t_{n+1})))}{\partial x^2},$$

then we have the following form of local growth in time of the spatial discretization error

$$\begin{aligned} \hat{est}_j(t_{n+1}) &= \triangle t \left[\left(\frac{(r+ \mid r \mid)(r-1)}{r(1+r)} \right) \frac{\partial V_j(t_{n+1})}{\partial x} \right] \\ &+ \ \triangle t \left[\frac{\triangle x^2}{4} \mu \frac{\partial^2 \tilde{\psi}(V_j(t_{n+1}))}{\partial x^2} \right]. \end{aligned} \tag{4.83}$$

It is evident that first term on the right hand side of above equation will be zero for r negative and and the source term is only active when $0 \ < \ V \ < \ 1$. Now using equations (4.46) and (4.48) the above equation can be written as

$$\hat{est}_j(t_{n+1}) = \triangle t \left[\triangle x \frac{(V_j(t_{n+1}))_{xx}}{2} + \frac{\triangle x^2}{4} \mu \frac{\partial^2 \tilde{\psi}(V_j(t_{n+1}))}{\partial x^2} \right]. \tag{4.84}$$

The *error balancing* approach is given by

$$\parallel \mathbf{le}(t_{n+1}) \parallel = \ \epsilon \parallel \hat{\mathbf{est}}(t_{n+1}) \parallel, \tag{4.85}$$

then using the value of $\hat{\mathbf{est}}(t_{n+1})$ from equation (4.84) the above equation can be written as

$$\parallel \hat{\mathbf{le}}(t_{n+1}) \parallel = \ \epsilon \triangle t \parallel \triangle x \frac{\mathbf{V}(t_{n+1})_{xx}}{2} + \frac{(\triangle x)^2}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \parallel . \tag{4.86}$$

The combination of equations (4.79) and (4.86) gives

$$\triangle t_{errbal} \ = \ \epsilon \frac{\parallel \triangle x \frac{\mathbf{V}(t_{n+1})_{xx}}{2} + \frac{(\triangle x)^2}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \parallel}{\parallel \frac{\partial^2 \mathbf{V}(t_{n+1})}{\partial t \partial x} + \mu \left(1 - 2\mathbf{V}(t_{n+1})\right) \frac{\partial \mathbf{V}(t_{n+1})}{\partial t} \parallel}, \tag{4.87}$$

and more precisely

$$\frac{\triangle t_{errbal}}{\triangle x} \ = \ \epsilon \frac{\parallel \frac{\mathbf{V}(t_{n+1})_{xx}}{2} + \frac{(\triangle x)}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \parallel}{\parallel \frac{\partial^2 \mathbf{V}(t_{n+1})}{\partial t \partial x} + \mu \left(1 - 2\mathbf{V}(t_{n+1})\right) \frac{\partial \mathbf{V}(t_{n+1})}{\partial t} \parallel}. \tag{4.88}$$

This equation can be modified by using the equation (4.57) to substitute for the time derivatives gives:

$$\frac{\triangle t_{errbal}}{\triangle x} \ = \ \epsilon \frac{\parallel \frac{\mathbf{V}_{xx}(t_{n+1})}{2} + \frac{(\triangle x)}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \parallel}{\parallel \frac{C_N}{\triangle x} \left(\frac{C_N \mathbf{V}(t_{n+1})}{\triangle x} + \mu \tilde{\psi}(\mathbf{V}(t_{n+1})) \right) + \mu \tilde{\psi}_V \left(\frac{C_N}{\triangle x} \mathbf{V}(t_{n+1}) + \mu \tilde{\psi}(\mathbf{V}(t_{n+1})) \right) \parallel}. \tag{4.89}$$

It is evident that when μ is very large then the terms in which the factor μ is appearing, will be dominant, because each $V_j(t_{n+1})$ lies in the interval [0,1]. So ignoring the term in which the factor μ is absent we have that

$$\mu \frac{\triangle t_{errbal}}{\triangle x} \approx \epsilon \frac{\| \frac{\triangle x}{4} \frac{\partial^2 \tilde{\psi}(\mathbf{V}(\mathbf{t_{n+1}})}{\partial x^2} \|}{\| \tilde{\psi}_V \tilde{\psi}(\mathbf{V}(t_{n+1})) \|}. \tag{4.90}$$

Hence by varying ϵ we are in fact controlling $\mu \frac{\triangle t}{\triangle x}$. From the above equation we see that the factor μ does not appear on the right hand side and $\tilde{\psi}$ is only a function of $\mathbf{V}(t_{n+1})$ and every component of $\mathbf{V}(t_{n+1})$ lies in [0, 1], so we conclude the right hand side is small as compared to μ. So for large μ and moderate $\triangle x$ the crude approximation is

$$\triangle t_{errbal} \approx \frac{1}{\mu}. \tag{4.91}$$

Our goal here is to find the numerical solution of equation (4.1) when the discontinuity presents present in the initial data and discontinuity must move with correct speed. It has been reported by Leveque and Yee [52] that insufficient spatial resolution may result in incorrect wave speed when μ is large. So with the implementation of equation (4.91) we shall derive the condition for spatial resolution when μ is large.

4.5.3 Stability

Now we discuss the issue of the stability when the LEPUS strategy is being implemented. The time step used when LEPUS is being implemented was explained in the previous Section and given as

$$\triangle t_{errbal} = \epsilon \frac{\| \triangle x \frac{\mathbf{V}_{xx}(t_{n+1})}{2} + \frac{(\triangle x)^2}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \|}{\| \frac{\partial^2 \mathbf{V}(t_{n+1})}{\partial t \partial x} + \mu \left(1 - 2\mathbf{V}(t_{n+1})\right) \frac{\partial \mathbf{V}(t_{n+1})}{\partial t} \|}. \tag{4.92}$$

The PDE to be solved is given by equation (4.1) and for this analysis we have assumed $\psi(V) = \mu V(1-V)$, where V is the numerical approximated solution given by the time integrator, and for $\frac{\partial \mathbf{V(t)}}{\partial t}$ we have the following form of equation (4.87).

$$\triangle t_{errbal} = \epsilon \frac{\| \triangle x \frac{\mathbf{V}_{xx}(t_{n+1})}{2} + \frac{(\triangle x)^2}{4} \mu \frac{\partial^2 \tilde{\psi}(\mathbf{V}(t_{n+1}))}{\partial x^2} \|}{\| \frac{\partial^2 \mathbf{V}(t_{n+1})}{\partial t \partial x} + \left(1 - 2\mathbf{V}(t_{n+1})\right) \left(\mu \frac{\partial \mathbf{V}(t_{n+1})}{\partial x} + \mu^2 \mathbf{V}(t_{n+1})(1 - \mathbf{V}(t_{n+1}))\right) \|}. \tag{4.93}$$

It is interesting when μ is very large, from equation (4.91) we have that $\triangle t_{errbal} \approx \frac{1}{\mu}$. The stability criteria for convergence of the iteration can be written as (see equation

(3.36))

$$\| \, [I - \triangle t_{errbal} \tilde{\gamma} J_s]^{-1} \triangle t_{errbal} \tilde{\gamma} J_f \, \| < \, r_c, \quad \text{where } r_c \, < \, 1, \tag{4.94}$$

and for the Backward Euler Method $\tilde{\gamma} \; = \; 1$,

$$\| \, [I - \triangle t_{errbal} J_s]^{-1} \triangle t_{errbal} J_f \, \| < \, r_c, \quad \text{where } r_c \, < \, 1. \tag{4.95}$$

In the above equation J_s is the Jacobian of the source term. It is clear from equation (4.58) that $J_s \; \approx \; -\mu \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}$ (for $\tilde{\psi}$ see equation (4.68)), then the above equation can be written as

$$\| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}]^{-1} \triangle t_{errbal} J_f \, \| < \, r_c, \text{ where } r_c \, < \, 1. \tag{4.96}$$

For large μ the time step is $\triangle t_{errbal} \;\; \approx \;\; \frac{1}{\mu}$ (see equation (4.91)), and multiplying both side of equation (4.96) with $\| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}] \, \|$ we have that

$$\| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}] \, \| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)}]^{-1} \triangle t_{errbal} J_f \, \| < \| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial V(t_n)}] \, \| \, r_c. \tag{4.97}$$

Using the identity

$$\| \, ab \, \| \leq \| \, a \, \| \| \, b \, \|, \tag{4.98}$$

the above equation can be written as

$$\| \, \triangle t_{errbal} J_f \, \| < \| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial V(t_n)}] \, \| \, r_c \qquad \text{where } \; r_c \, < \, 1. \tag{4.99}$$

Now the entries of the diagonal matrix $[I \; + \; \frac{\partial \tilde{\psi}(\mathbf{V}(t_n)}{\partial \mathbf{V}(t_n)}]$ can be written as

$$e_{ij} \; = \; \begin{cases} 1 + (1 - 2V_j(t_n)) \; = \; 2(1 - V_j(t_n)) \text{ if } i = j \\ 0 \quad \text{otherwise} \end{cases}$$

because the equations are not coupled. The expression $\left[\frac{\partial \tilde{\psi}(\mathbf{V}(t_n)}{\partial \mathbf{V}(t_n)} \right]$ is given by equation (4.69). It is evident that the maximum value of e_{ij} will be 2 because the minimum value of V is zero, so that the quantity $\| \, I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)} \, \|$ will be less than or equal to 2 if we used the max norm. Again equation (4.58) reveals that $J_f \; = \; \frac{C_N}{\triangle x}$ so that the stability criteria as given by equation (4.99) has the following form

$$\| \, \triangle t_{errbal} \frac{C_N}{\triangle x} \, \| < r, \tag{4.100}$$

where r is defined by $\| \, [I + \frac{\partial \tilde{\psi}(\mathbf{V}(t_n))}{\partial \mathbf{V}(t_n)} \, \| \, r_c$ and $\triangle t_{errbal} \;\; \approx \;\; \frac{1}{\mu}$ (equation (4.91)) hence we have that

$$\| \, \frac{C_N}{\triangle x \mu} \, \| < \, r, \tag{4.101}$$

which gives

$$\triangle x \;\approx\; O(\frac{1}{\mu}). \qquad (4.102)$$

4.5.4 Calculated Wave Speed

An important feature of solving the Leveque and Yee problem [52] is that the numerical solution may move with an incorrect wave speed. The computed wave speed in time step t_n is given by (see for example [52])

$$\begin{aligned} \text{wave speed} &= \frac{\triangle x}{\triangle t}\sum_j (V_j(t_n) - V_j(t_{n-1})) \qquad (4.103) \\ &= \frac{\triangle x}{\triangle t}\alpha \;\text{ where }\; \alpha \;=\; \sum_j (V_j(t_n) - V_j(t_{n-1})), \end{aligned}$$

where $V_j(t_n)$ and $V_j(t_{n-1})$ represent the computed solution at the nth and (n-1)th time step and jth grid point. To have an idea of the time step and mesh size so as to move the front with correct speed, we have used the initial data given as

$$u(x,0) = u_0(x) = \begin{cases} u_L \;=\; 1 & \text{if } x \le 0.2 \\ u_R \;=\; 0 & \text{if } x > 0.2. \end{cases}$$

The initial data shows that the initial discontinuity is at $x = 0.2$. Let S be the speed of discontinuity and for this problem $S = 1$. Then the relative error in the speed is given by (see [61])

$$\text{re} = \frac{S_{numerical} - S_{exact}}{S_{exact}}, \qquad (4.104)$$

where $S_{numerical}$ is the average wave speed given below

$$\text{Average speed} = \frac{\triangle x}{t_n - t_0}\left(\sum_j V_j(t_n) - \sum_j V_j(t_0)\right), \qquad (4.105)$$

where $\triangle x$ is the mesh size. Figure (4.1) displays the relative error in speed for local error per step with RTOL = 0.01. Here we have taken the CFL number $c = .3$ and $\triangle t = \frac{.2}{2^i}$ $i = 5,...7$. In order to calculate the mesh we have used the identity $c = \frac{\triangle t}{\triangle x}$ (the CFL number) and this has been assumed to be fixed with the value 0.3 in the calculation of the mesh size. The pre-multiplication factor μ of the source term as given by equation (4.67) has been taken as 1000.

It is clear from Figure (4.1) that in the regime $\frac{1}{\mu\triangle t} \geq .3$ $(\triangle t \leq \frac{.3}{\mu})$ the relative error in speed is nearly 20% and after that it is still decreasing and correspondingly

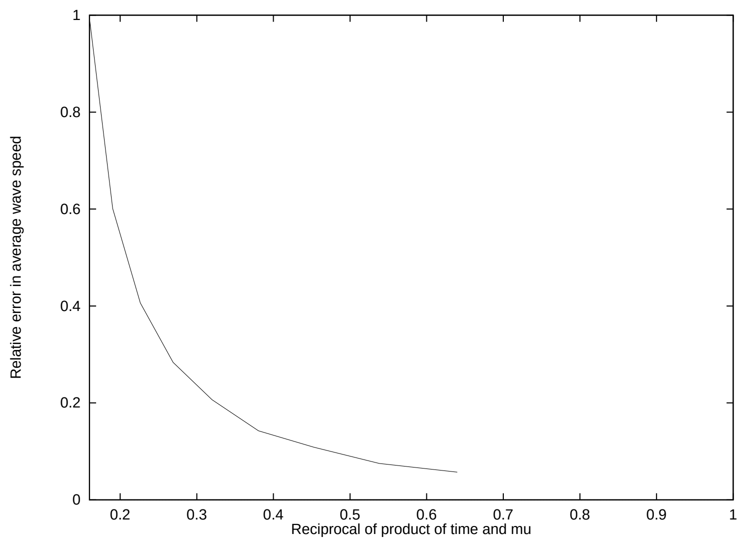


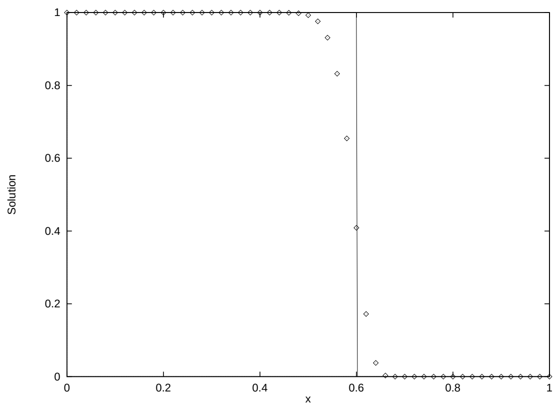
Figure 4.1: Variation of relative error in average speed with the reciprocal of the stiffness factor $\triangle t\mu$ using the LEPS strategy.

the mesh size will be $\frac{\triangle t}{c}$ where c is the CFL number. These observations reveal that the step size and the mesh size should be O($\frac{1}{\mu}$), to avoid spurious solutions, which is the same observation as we derived analytically in Section (4.5.4).
For the comparison to Leveque et, al [52] results we have taken initial data

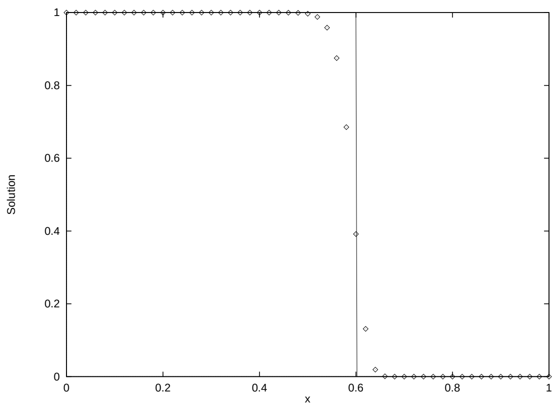
$$u(x,0) = u_0(x) = \begin{cases} u_L &= 1 \quad \text{if } x \leq 0.3 \\ u_R &= 0 \quad \text{if } x > 0.3, \end{cases}$$

and $\triangle x$ =0.02 and fixed time step $\triangle t$= 0.015. The product of time step $\triangle t$ and the reaction rate μ determines the stiffness of the system. Figure(4.2) shows the comparison of the computed solution and exact solution at $t = 0.3$ for μ =1, 10, 100, and 1000 ($\triangle t\mu$=0.015, 0.15, 1.5 and 15) respectively, with the strategy of controlling the local error per step.

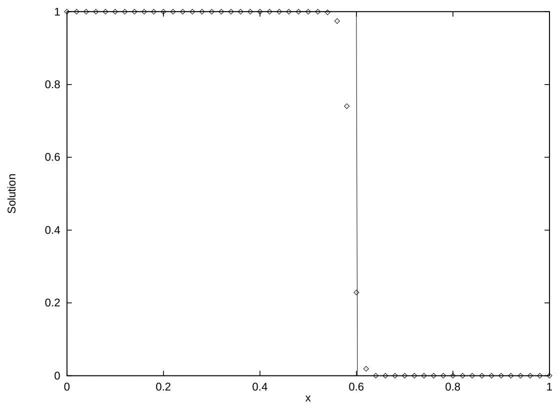
It is evident from Figure (4.2) that for smaller $\triangle t\mu$ with the local error control strategy the front is moving with correct speed. In the case when $\triangle t\mu = 15$, the discontinuity has remained at x = 0.3 and when we switched to another quadrature rule (the trapezoidal rule), a large undershoot and overshoot were seen . The same numerical experiment has been performed with even smaller step sizes and similar results have been obtained. This incorrect propagation of the discontinuity is due to the lack of proper spatial resolution. In the previous Section we have shown that $\triangle x \approx \frac{1}{\mu}$ to avoid this wrong propagation of the discontinuity.



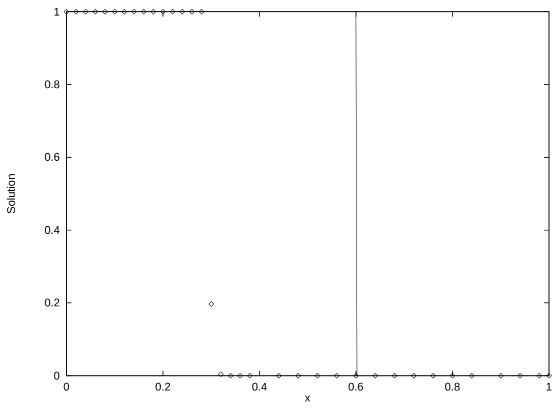
Solution with $\triangle t\mu = 0.015$



Solution with $\triangle t\mu = 0.15$



Solution with $\triangle t\mu = 1.5$



Solution with $\triangle t\mu = 15$

Figure 4.2: Comparison of true solution(line) and numerical solution(dots) using the LEPS, for 0.01 relative tolerance and 1×10^{-5} absolute tolerance.

4.6 Local Grid Refinement

In the previous Section it was shown that the front is moving with the wrong speed, due to lack of proper spatial resolution. This is not surprising , because the source of difficulty is the discontinuity in the initial data and much finer grid is needed in the vicinity of the discontinuity. Our analysis shows that spatial resolution is as important as the temporal resolution. One solution to such problems is to deploy a method that is capable of essentially increasing the spatial resolution rather than excessive refinement of the overall grid.

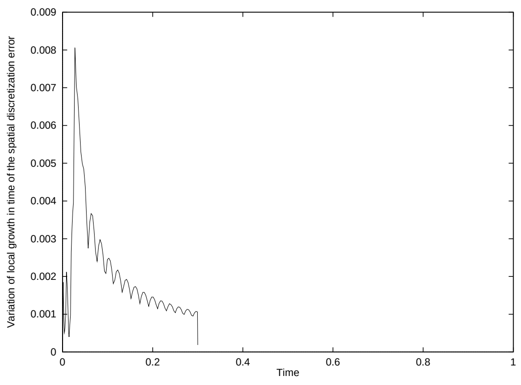
When the error balancing approach described in the Section (4.3) is applied Figure (4.3) displays the local growth in time of the spatial discretization error as given by equation (4.53) for ($\triangle t\mu = 0.015,\ 0.15,\ 1.5,$ and 15) and it is evident that the local growth in time of spatial discretization error is growing with increasing $\triangle t\mu$ which is evident from the equation (4.53). The trend of the local growth in time of spatial discretization error corresponding to $\triangle t\mu = 0.015$ and 0.15 is identical and decreasing in time after an initial increase. In contrast for $\triangle t\mu = 1.5$,i.e., $\mu = 100$ the error has increased rapidly due to increasing stiffness of the source term (see equation (4.53)) and stays at a higher level than the previous two cases. The wiggles are more prominent as compared to previous cases, which may be due to insufficient spatial resolution. The case corresponding to $\triangle t\mu = 15$ of Figure (4.3) shows that the error is increasing and interestingly for $\triangle t\mu = 15$ is constant because the discontinuity does not move. From this it seams that much finer grid is needed to ensure the correct movement of the discontinuity.

The numerical experiments have revealed that this error is growing sharply near the discontinuity, which is only due to the lack of proper spatial resolution. It implies that higher resolution is only needed in that part of domain where discontinuity exists.

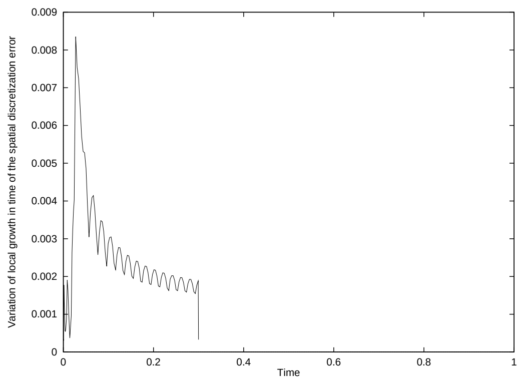
This leads to the concept of local grid refinement, and with the help of the error balancing approach described in Section (4.3) it is possible to create a new refined grid directly surrounding of the location of the source. The local growth in time of the spatial discretization error measured by the error balancing approach has been taken as the remeshing monitor function. The mesh cell is refined if the monitor function is greater than a specified limit.

4.6.1 Spatial remeshing using the MONITOR routine

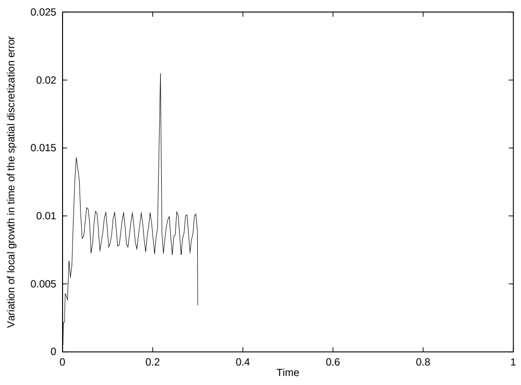
The SPRINT package (Software for Problem IN Time), is a state-of-the-art computer program, which deals with the numerical solution of mathematical models that involve mixed system of time-dependent algebraic, ordinary and partial differential equations (ODEs and PDEs) see [8]. The important property of SPRINT is that it has the ability to handle both discrete and continuous remeshing schemes. After the success of each time-step of the integrator a routine, generic name MONI-



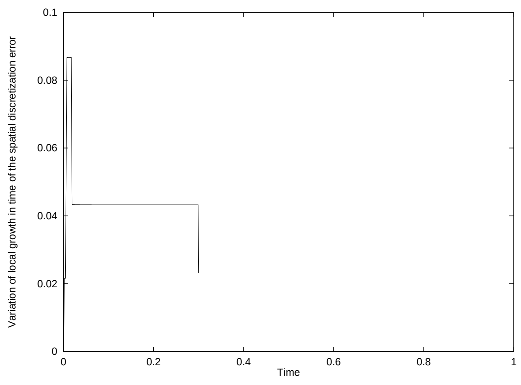
Local growth in time of spatial error for $\triangle t\mu = 0.015$



Local growth in time of spatial error for $\triangle t\mu = 0.15$



Local growth in time of spatial error for $\triangle t\mu = 1.5$



Local growth in time of spatial error for $\triangle t\mu = 15$

Figure 4.3: Local growth in time of the spatial discretization error for different value of stiffness factor $\triangle t\mu$.

TOR, is called which has the flexibility of performing various tasks. The key feature of the MONITOR routine is that it has the power to access the whole of the non-linear solver in SPRINT and has been designed for tasks such as ODEs global error estimates, discrete time remeshing, etc.

In discrete remeshing processes a new mesh is created at certain times in integration, the solution and its time derivatives are interpolated onto the new mesh and the integration is continued. For this purpose we have modified the SPDIFF routine as developed by Berzins and Fuzerland [31]. The key difference is that we have used a new monitor function based on local growth in time of the spatial discretization error (see Section 4.3.1). The remesh routine applies the ideas of [14] to construct a new mesh at the current time step, i.e., bisecting the mesh cell if the monitor function is too large or combining the two mesh points if the monitor function at that point is well below the required value. In the numerical experiments here, the remeshing routine is called on every second time step.

When a decision is made that the new mesh is needed and the remesh routine has evaluated a new mesh, then the next task performed by the MONITOR routine is to evaluate the solution, its time derivative and any other higher time derivatives used by the ODEs integrator on the new mesh, by using cubic spline interpolation. Then the time integration attempts to continue directly using the stepsize and order determined at the end of the step prior to remeshing. The case in which the number of mesh points has changed, and hence which will change the size of the DAE system being integrated in time makes it necessary to calculate the Jacobian matrix before integration can continue.

Initially we start with 26 points and when the error was larger than the specified limit then we divide the corresponding cell into two with a maximum 75 points being allowed. Figures (4.4) and (4.5) reveal that the front is moving with the correct

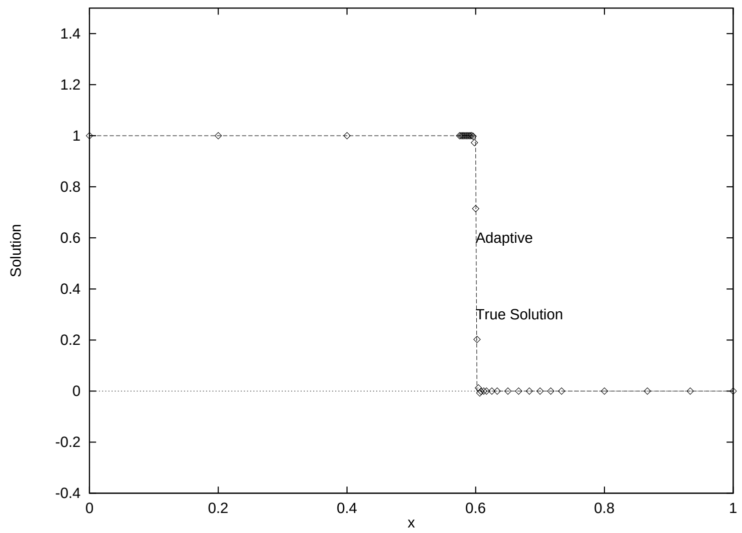


Figure 4.4: Comparison of true solution(lines) and numerical solution(dots) using grid refinement technique at time equal to 0.3.

speed. The statistics in Tables (4.1) and (4.3) reveal that the Error Balancing approach automatically gives the correct level of temporal error while monitoring efficiency as compared to the LEPS approach.

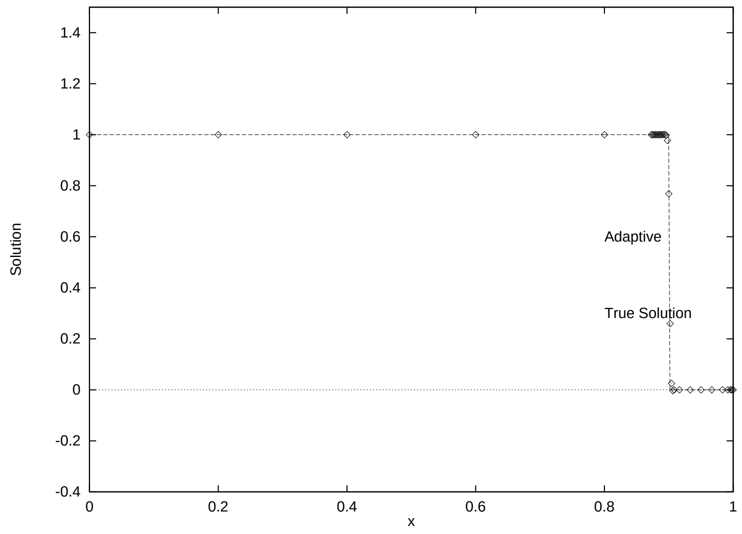


Figure 4.5: Comparison of true solution (lines) and numerical solution(dots) with grid refinement technique at time equal to 0.6.

4.7 2D Problems

In this Section we will focus on a 2D version of Leveque and Yee ([52]) in order to have the evidence of the accuracy, robustness, reliability of our new error control strategy (i.e. local error per unit step). The Leveque and Yee [52] problem in two space dimension takes the following form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = -\psi(u) \qquad t \in [0, t_f] \; and \; (x, y) \in \Omega, \tag{4.106}$$

where u may be vector of dependent variables, but here we are assuming it to be a scalar. As usual the flux functions f and g contains the advective terms while the source term $\psi(u)$ arises from the reactive term into PDEs. For the spatial discretisation we have adopted the finite volume approach as described in Section (2.9) i.e,

$$\frac{dU_{i,j}}{dt} + \frac{[f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}]}{\triangle x} + \frac{[g_{j+\frac{1}{2}} - g_{j-\frac{1}{2}}]}{\triangle y} = -\phi(U_{i,j}(x, y)). \tag{4.107}$$

The next task is to find the convective fluxes midway along the edge of the volume element, which involves the approximate solution of the four one dimensional Riemann problems in the direction of the normals to edges of the volume element. To find the convective flux $f_{i+\frac{1}{2},j}$ and $f_{i-\frac{1}{2},j}$ we have adopted a similar procedure to that given in [7] such that

$$\begin{aligned} f_{i+\frac{1}{2},j} &= \hat{f}(x_{i+\frac{1}{2}}, y_j, t, U^x_{i,j}(x_{i+\frac{1}{2}}, y_j, t), U^x_{i+1}(x_{i+\frac{1}{2}}, y_j, t)), \\ f_{i-\frac{1}{2},j} &= \hat{f}(x_{i-\frac{1}{2}}, y_j, t, U^x_{i-1,j}(x_{i-\frac{1}{2}}, y_j, t), U^x_i(x_{i-\frac{1}{2}}, y_j, t)). \end{aligned}$$

In the above equation $U^x_{i,j}(x,y,t)$ represents the x-dimensional upwind interpolant from cell i,j, and is being estimated at the midpoint of the edge, and the $\hat{f}$ in fact is given by the solution of Riemann problem with $U^x_{i,j}(x,y,t)$ and $U^x_{i+1,j}(x,y,t)$ as the discontinuous values on each side of the edge. By a similar procedure we can estimate the convective fluxes along y-direction $g_{i,j+\frac{1}{2}}$ and $g_{i,j-\frac{1}{2}}$ as follows

$$\begin{aligned} g_{i,j+\frac{1}{2}} &= \hat{g}(x_i, y_{j+\frac{1}{2}}, t, U^y_{i,j}(x_i, y_{j+\frac{1}{2}}, t), U^y_{i,j+1}(x_i, y_{j+\frac{1}{2}}, t)), \\ g_{i,j-\frac{1}{2}} &= \hat{g}(x_i, y_{j-\frac{1}{2}}, t, U^y_{i,j-1}(x_i, y_{j-\frac{1}{2}}, t), U^y_{i,j}(x_i, y_{j-\frac{1}{2}}, t)), \end{aligned}$$

where $U^y_{i,j}(x,y,t)$ represents the dimensional upwind interpolant estimated at the mid point (x, y, t) and the function $\hat{g}$ is defined with the help of the solution of a Riemann problem with $U^y_{i,j}(x,y,t)$ and $U^y_{i,j+1}(x,y,t)$ as the discontinuous values on each side of the edge.

As mentioned in [7], limited combination of these interpolants gives more accurate estimates of solution values on the edge given as

$$\begin{aligned} U^x_{i-1,j}(x_{i-\frac{1}{2}}, y_i, t) &= U_{i-1,j}(t) + \tfrac{\Delta x}{2} \tfrac{U_{i-1,j}(t) - U_{i-2,j}(t)}{\Delta x} B(r^x_{i-1}, 1), \\ U^x_{i,j}(x_{i-\frac{1}{2}}, y_i, t) &= U_{i,j}(t) - \tfrac{\Delta x}{2} \tfrac{U_{i+1,j}(t) - U_{i,j}(t)}{\Delta x} B(\tfrac{1}{r^x_i}, 1), \end{aligned}$$

where the ratio r^x_i is defined by

$$r^x_i = \frac{U_{i+1,j}(t) - U_{i,j}(t)}{U_{i,j}(t) - U_{i-1,j}(t)}, \tag{4.108}$$

and it is further mentioned in Berzins [7] that different limiter functions B(.,.) give rise to different spatial accuracies. With this implementation of the 2D spatial discretization and appropriate boundary conditions to the equation (4.106) we get the following system of ODEs

$$\dot{\mathbf{U}} = \mathbf{F}_K(t, \mathbf{U}(t)), \tag{4.109}$$

where the K-dimensional vector $K = N \times N$ is given by

$$[\mathbf{U}(t)]_m = U_{i,j}(t), \; m = (i-1) \times N + j, \; i = 1, ..., N, \; j = 1, ..., N. \tag{4.110}$$

The next task is to find the solution of the above equation for which we will apply the local error per unit step control strategy rather than the local error per step control strategy to check the performance of local error per unit step control strategy on a 2D problem.

4.8 Numerical Results

Problem 1 First we have solved the 1D Leveque and Yee problem [52] with evenly spaced, meshes having 21, 41, 81 and 161 points as well as the adaptive mesh described earlier. The test initial data is given by

$$u(x,0) = u_0(x) = \begin{cases} u_L &= 1 \quad \text{if } x \le 0.1 \\ u_R &= 0 \quad \text{if } x > 0.1. \end{cases}$$

The length of domain is $L = 1$ and time t_f=1. As regards the boundary condition, at origin we have solved an ordinary differential equation of the form

$$u_t \;=\; -\psi(u). \tag{4.111}$$

On the right hand side an outflow boundary condition has been applied given by

$$f_{N+\frac{1}{2}} \;=\; f_{N-\frac{1}{2}}, \tag{4.112}$$

where N is the number of mesh cells.

Problem 2 Then we solved the 2D Leveque and Yee problem with fixed mesh size. The initial data is given below

$$u(x,y,0) = u_0(x,y) = \begin{cases} u_L &= 1 \quad \text{if } 0.5(x+y) \le 0.1 \\ u_R &= 0 \quad \text{if } 0.5(x+y) > 0.1. \end{cases}$$

Again the simulation time is 1 and domain $\Omega = [0,1] \times [0,1]$. The next task is to implement appropriate boundary conditions. So at the origin we have solved the an ordinary differential equation as given in Problem 1. Along x-axis we have taken $\frac{\partial g(u)}{\partial y} = 0$ and on y-axis we assumed $\frac{\partial f(u)}{\partial x} = 0$. On the right boundary we have applied the outflow boundary condition given by

$$f_{N+\frac{1}{2},j} \;=\; f_{N-\frac{1}{2},j}, \tag{4.113}$$

and similarly the top boundary has been treated.

4.8.1 Results Discussion

Two different kinds of error control strategies have been used within the integration routine:

Method A: The LEPS strategy that is controlling the local error $\mathbf{le}(t_{n+1})$ so that

$$\| \frac{\mathbf{le}(t_{n+1})}{RTOL \mid \mathbf{V}(t_n) \mid + ATOL} \| < \; 1, \tag{4.114}$$

Method B: The LEPUS strategy that is controlling the local error $\mathbf{le}(t_{n+1})$ so that

$$\| \mathbf{le}(t_{n+1}) \| = \epsilon \| \hat{\mathbf{est}}(t_{n+1}) \|, \tag{4.115}$$

where $\hat{\mathbf{est}}(t_{n+1})$ is the local growth in time of the spatial discretisation error and is given by Section (4.3.1). Before comparing the efficiency of these local control strategies, the choice of the parameter ϵ should be discussed.

The vital fact in the selection of this parameter is that local growth in the spatial discretization error should dominate the temporal error and work needed while implementing this technique should be minimum. Obviously the larger the value of ϵ the fewer ODE time steps there will be, and the smaller the value of ϵ the more steps there will be (see for example Figure (4.6)). We have also plotted CFL number by varying balancing factor ϵ (see Figure (4.9)) and it is evident that CFL number is arising with increasing balancing factor.

It is also clear that the CFL number increases sharply near the time $t = 1$, due to the sudden rise of local growth in the spatial discretization error at this time.

Also equation (4.115) shows that when the local growth in time of the spatial discretization error is larger, the local error test being made in the solver satisfied, but this may result in the convergence failure of the non-linear solver, which will force to the reduction of the time step. That is why the the graphs in Figure (4.9) shows sudden fall after it has increased.

Additionally, Figure (4.9) shows that when $\epsilon > 0.5$ the CFL number increases more sharply and when $\epsilon < 0.3$ the code is taking more steps, which is clear from Figure 4.6. Hence any value of ϵ between 0.3 and 0.5 can be used in the calculation for the mildly stiff problem, the factor may have to be reduced in case of highly stiff problem such as combustion.

In Figures(4.8) and (4.7) we have plotted the norm of the local growth in time of the spatial discretization error and is increasing sharply near time t equal to 1. The reason is that at t equal to 1 the solution values are not exactly equal to one, and the source term becomes active and we get a sudden rise in the local growth in time of the spatial discretization error.

The following notation has been used to represent the result:

- **NPTS** = The number of points used in the spatial discretisation,
- **Nsteps** = The number of integration steps used by the integrator,

- **ATOL** = Absolute error tolerances,
- ϵ =balancing factor,
- **RTOL** = Relative error tolerance,
- **G-S** = The number of Gauss Seidel iterations.

The numerical results obtained for 1D problem are displayed in Figure(4.10) with Method B (see equation (4.115)) and in Figure(4.11) with method A (see equation (4.114)) and comparison has been made with exact solution.

Figure (4.10) shows the comparison of exact and numerical solution by using the method B, while Figure (4.11) displays the comparison when the method A is being implemented for (RTOL = 0.01) respectively at t=0.5 and its maximum solution at t=1 is 1. From these Figures it is evident that the results are of comparable accuracy with both strategy for $\mu = 10$.

In order to have more information about the accuracy of the both strategies, we have presented the results at t_f=1 in Table (4.2) using different μ, different number of points, and different RTOL when the LEPS strategy is used. It is evident that for $\mu = 10$ and NPTS=161, the results with LEPUS strategy is as accurate as RTOL=0.001 with the LEPS strategy, and for $\mu = 100$ again the results are more accurate than the LEPS with RTOL=0.01 and the number of steps has been reduced by almost 50%. The reason for the reduction of steps with LEPUS strategy is that by increasing μ the local growth in time of the spatial discretization error has increased, consequently the local growth in time of the spatial discretization error dominates the temporal error. Almost the same behaviour can be observed when the number of points has been reduced to 81.

The dimensionless parameters the mesh ratio $c = \frac{\triangle t}{\triangle x}$ and product $\triangle t\mu$ of the time step and reaction rate play a vital role in the performance method and the stiffness of the system is determined by the product $\triangle t\mu$.

Figure (4.2) shows there is some discrepancy in the location of the discontinuity, when $\triangle t\mu = 1.5$ and the situation become worse for $\triangle\mu = 15$, because the discontinuity has not moved from $x = 0.3$. These results have been obtained with $\triangle t = 0.015$, and experiments have been performed even with smaller time steps but no improvement was achieved in the results.

Methods	NPTS	ATOL or ϵ	RTOL	Nsteps	Fun	Jac	G-S
A	21	10^{-5}	10^{-1}	154	650	43	794
		10^{-5}	10^{-2}	254	970	52	1196
		10^{-5}	10^{-3}	471	1720	77	2140
B	21	0.3		155	605	33	1068
A	41	10^{-5}	10^{-1}	255	1031	57	1292
		10^{-5}	10^{-2}	454	1736	90	2146
		10^{-5}	10^{-3}	808	2728	82	3472
B	41	0.3		273	1022	51	1822
A	81	10^{-5}	10^{-1}	462	2068	130	2594
		10^{-5}	10^{-2}	814	3025	137	2626
		10^{-5}	10^{-3}	1316	4298	100	5534
B	81	0.3		600	2011	58	3782
A	161	10^{-5}	10^{-1}	820	3754	229	4772
		10^{-5}	10^{-2}	1384	5135	222	6460
		10^{-5}	10^{-3}	2467	8326	240	10634
B	161	0.3		1242	4096	107	7756

Table 4.1: Results of 1D Leveque and Yee problem with uniform grid using the LEPUS(B) and LEPS(A) control strategies.

NPTS	μ	Method	ATOL or ϵ	RTOL	NSTEPS	Solution at t_f=1
161	10	A	1×10^{-5}	0.1	820	1.0154
			1×10^{-5}	0.01	1384	0.9997
			1×10^{-5}	0.001	2467	0.9997
		B	0.3		1242	0.9999
	100	A	1×10^{-5}	0.1	881	0.9815
			1×10^{-5}	0.01	1727	0.9987
			1×10^{-5}	0.001	2966	1.0000
		B	0.3		798	0.9999
81	10	A	1×10^{-5}	0.1	462	1.0062
			1×10^{-5}	0.01	814	0.9983
			1×10^{-5}	0.001	1316	0.9970
		B	0.3		600	0.9998
	100	A	1×10^{-5}	0.1	485	1.0095
			1×10^{-5}	0.01	923	0.9980
			1×10^{-5}	0.001	1831	1.0002
		B	0.3		426	0.9999

Table 4.2: Table showing the accuracy of the numerical solution of 1D Leveque and Yee problem with Both strategies at time =1 by using different number of points with different value of μ.

This implies that the spatial resolution is important for having correct propagation of the reaction front. We conclude that the controlling of local error per step (see equation (4.114)) gives an accurate speed propagation unless insufficient spatial resolution has been used.

Methods	NPTS	ATOL or ϵ	RTOL	Nsteps	Fun	Jac	G-S
A	10×10	1×10^{-4}	1×10^{-1}	199	1133	135	1320
	10×10	1×10^{-4}	1×10^{-2}	187	1084	126	1284
	10×10	1×10^{-4}	1×10^{-3}	191	1030	106	1246
B	10×10	0.3		205	1113	120	1730
A	20×20	1×10^{-4}	1×10^{-1}	414	2547	305	3008
	20×20	1×10^{-4}	1×10^{-2}	497	3077	369	3642
	20×20	1×10^{-4}	1×10^{-3}	448	2586	290	3078
B	20×20	0.3		478	2803	324	4274
A	30×30	1×10^{-4}	1×10^{-1}	722	4356	517	5122
	30×30	1×10^{-4}	1×10^{-2}	809	5011	598	5944
	30×30	1×10^{-4}	1×10^{-3}	811	4857	565	5756
B	40×40	0.3		650	3840	446	5838
A	40×40	1×10^{-4}	1×10^{-1}	1359	8568	1041	10120
	40×40	1×10^{-4}	1×10^{-2}	1146	7203	870	8528
	40×40	1×10^{-4}	1×10^{-3}	1112	6826	800	8130
B	40×40	0.3		1230	7399	850	11260

Table 4.3: Results of 2D Leveque and Yee problem with uniform grid using the LEPUS(B) and LEPS(A) control strategies.

We have explained in Section (4.5.4) that for correct movement of the discontinuity, it is essential that $\triangle t\mu \leq 1$, hence when $\triangle t\mu = 15$, a much finer grid is needed to model the correct movement of the front.

Hence it is better to refine the grid locally rather than globally, because this is a more efficient way to tackle the problem given that substantial refinement is necessary to obtain reasonable results.

For this purpose we need some sort of monitor function, so that the decision can be made where to refine. The commonly used monitor function is the spatial derivative which tend to infinity around a shock [62] as the mesh is refined. We have introduced a new monitor function based upon the local growth in time spatial error. The local growth in time spatial discretization error (see equation (4.53)) is increasing with the increasing stiffness of the source (see Figure (4.3)) and has been successfully implemented for the grid refinement (see Section (4.6)).

In order to check the reliability of the new control strategy LEPUS equation (4.115) on 2D problem the comparison has been made with the LEPS control strategy by using different grids and RTOL but fixed ATOL. Figures (4.12) and (4.13) show the results with LEPUS control strategy on 31×31 and 41×41 respectively at time equal to 0.6. It is evident that quality of results are improving by increasing the mesh resolution.

Figures (4.14) and (4.15) show the results with the LEPS on 31×31 and 41×41 grids respectively by taking RTOL=0.01 and ATOL=10^{-4}. It is evident there are small oscillations on coarser grid. We have also presented results in Figures (4.16) and (4.15) with same ATOL and RTOL=0.001 at time equal to 0.6, which shows that accuracy have increased by decreasing RTOL.

Comparison of results obtained with LEPUS and LEPS show that the results obtained with LEPUS are as accurate as obtained with LEPS using RTOL=0.001 and ATOL=10^{-4}.

Table (4.3) shows that when method A(LEPS) is being implemented at some grids with large RTOL values the code takes more steps as compared to the tighter tolerance. This appears to be happening due to convergence failures in the non-linear solver.

The efficiency of both strategies can be seen from Tables (4.1) and (4.3), which shows that it is worth using the local error per unit step control strategy rather than local error per step control strategy. Even though a modest tolerance has been used for the LEPS, (see equation (4.114)) strategy, the number of steps are comparable to the LEPUS equation (4.115) strategies thus showing the effectiveness of the latter approach.

4.9 Conclusion

The method of lines has been used to solve 1D and 2D Leveque and Yee problem [52]. The second order upwind together with the van Leer limiter [7] has been used as the spatial discretization and the theta method has been used as the time integrator [9]

In most existing software based upon the the method of lines, the standard procedure is to control the local time error per step [40] with respect to a supplied tolerance. It is difficult for the user to select a tolerance which is related to the spatial discretization error. Also the controlling of local error per step does not always guarantee equivalent control of the global time error. This implies that this technique is not ideal because a clear relationship can be found between the supplied tolerance and the global error.

Keeping in view these arguments we have derived a new control strategy (LEPUS) based upon the error balancing approach [7]. The underlying idea is that first we evaluate the primary solution by using an upwind scheme and quadrature rule for the source term integration, and then the secondary solution is evaluated with a less accurate upwind scheme and a different quadrature rule .

It is the difference between two solutions, the local growth in time of the spatial discretization error, that is used as time tolerance. It should be noted that this only estimates the local growth in time of the spatial discretization error in the less accurate solution. The statistic in Tables (4.1) and (4.3) have shown that the new control strategy works well in combination with the Gauss-Seidel iteration.

We have used the Gauss-Seidel iterative method because when solving big atmospheric problems storage is the restrictive factor. The storage requirement is reduced considerably by implementation of the Gauss-Seidel method [7] as compared to using conventional linear algebra solvers. The reduction of storage is the main reason for the popularity of the Gauss-Seidel method, for large systems of advection-reaction equations arising in the modelling of atmospheric pollution problems.

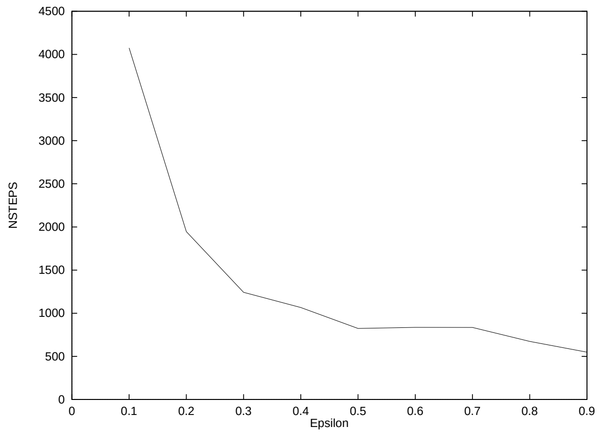


Figure 4.6: Graph showing relationship between number of steps and balancing factor ϵ, for 161 points and the source term factor equal to 10.

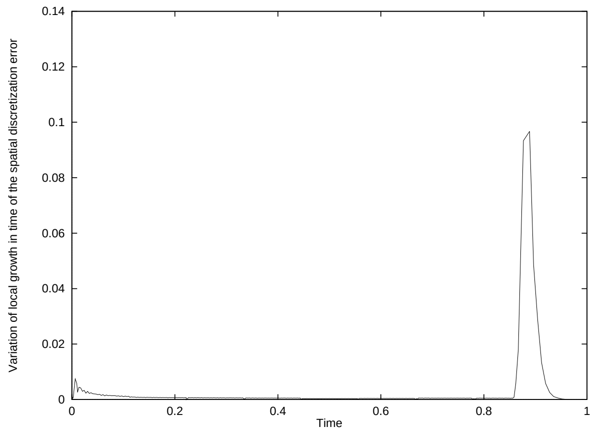


Figure 4.7: Variation of norm of local growth in time of the spatial discretization error using LEPUS strategy, 161 points, balancing factor 0.3 and the source term equal to 10.

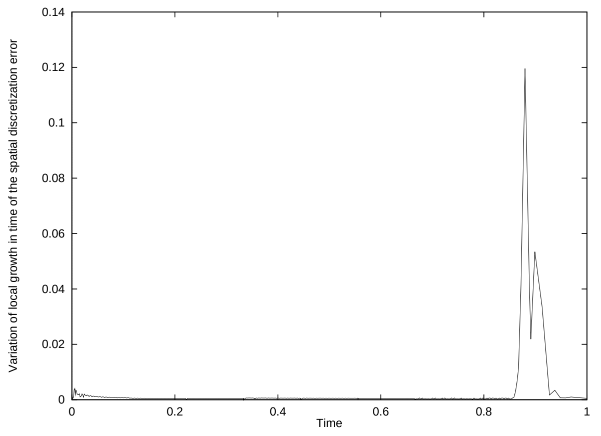
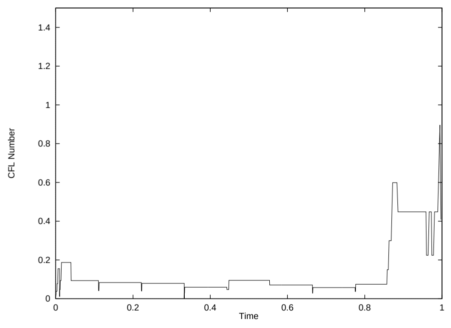
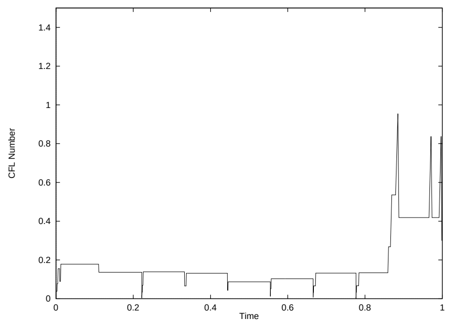


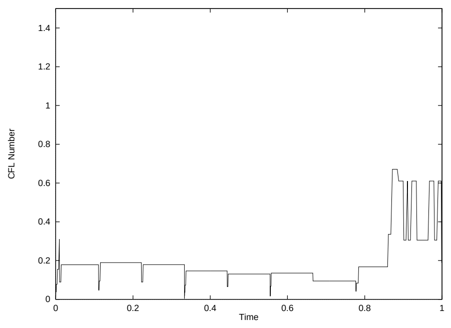
Figure 4.8: Variation of norm of local growth in time of the spatial discretization error using the LEPS strategy for 161 number of points, 0.01 relative tolerance, 1×10^{-5} absolute tolerance and the source term factor equal to 10.



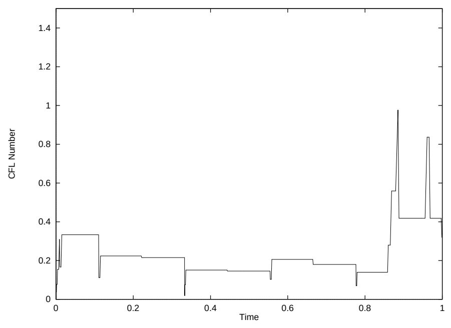
CFL number with $\epsilon = 0.2$



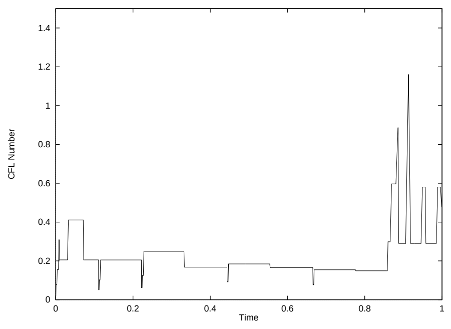
CFL number with $\epsilon = 0.3$



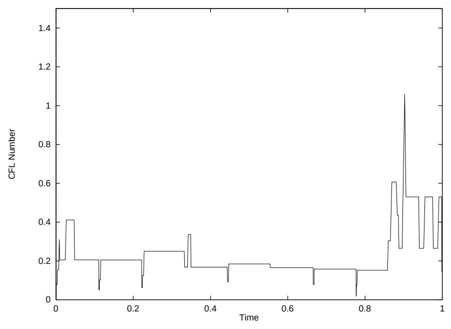
CFL number with $\epsilon = 0.4$



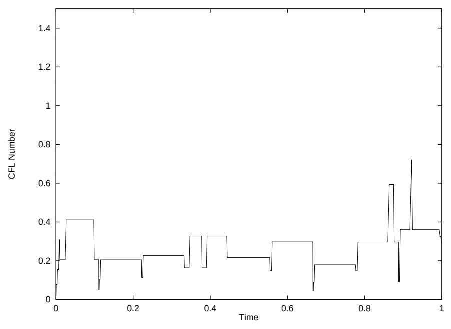
CFL number with $\epsilon = 0.5$



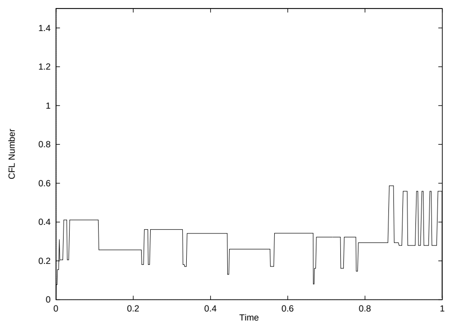
CFL number with $\epsilon = 0.6$



CFL number with $\epsilon = 0.7$

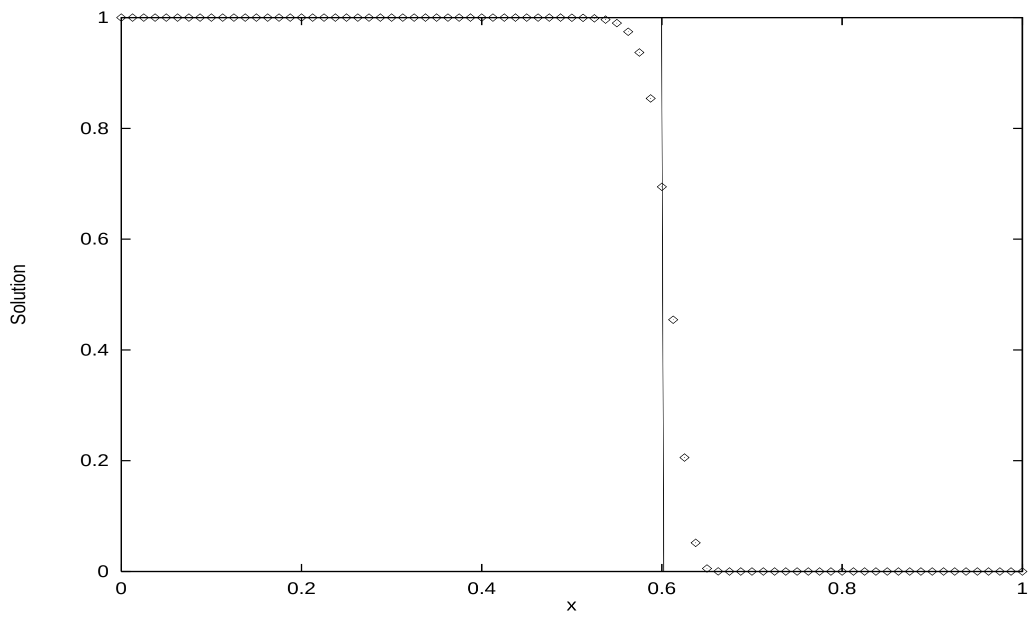


CFL number with $\epsilon = 0.8$



CFL number with $\epsilon = 0.9$

Figure 4.9: Variation of CFL number with balancing factor ϵ.



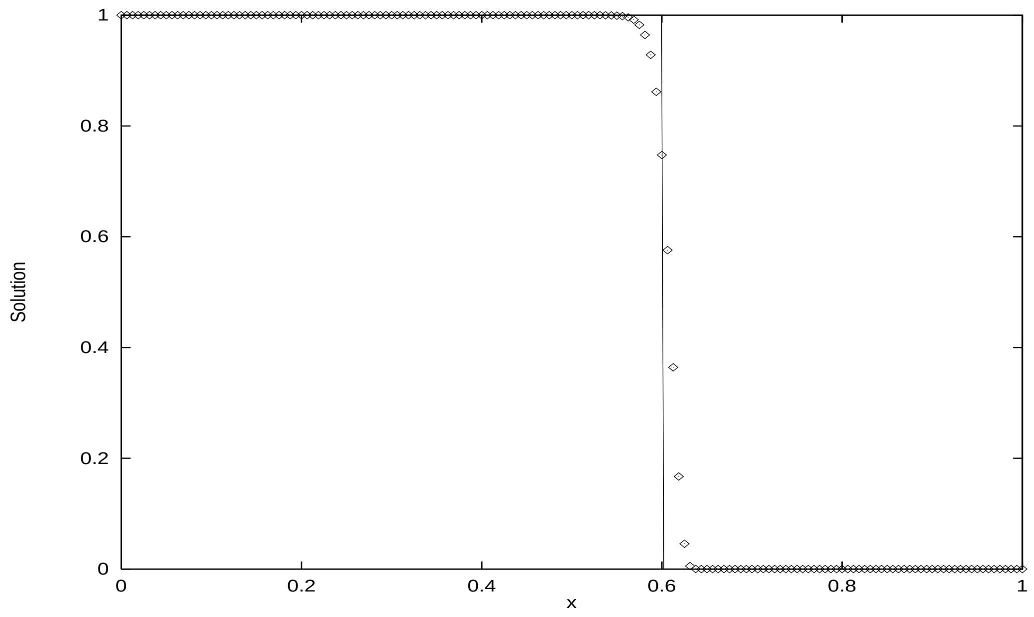
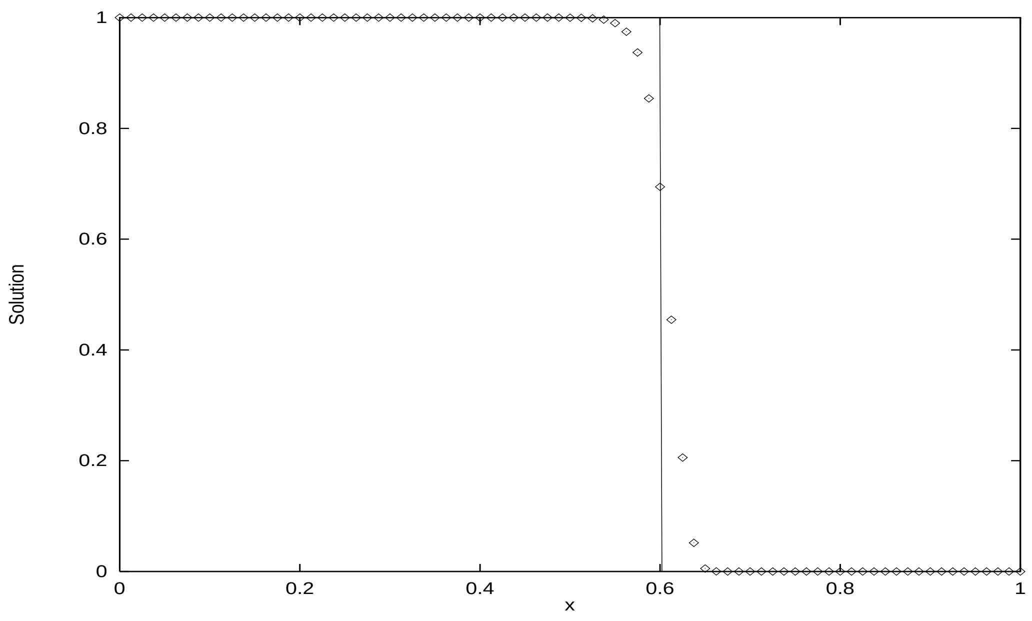


Figure 4.10: The comparison of true (line) and numerical(dot) solutions in 1D Leveque and Yee problem using LEPUS control strategy a uniform mesh of 81 points (top), 161 points(bottom) and balancing factor equal to 0.3. at time=0.5



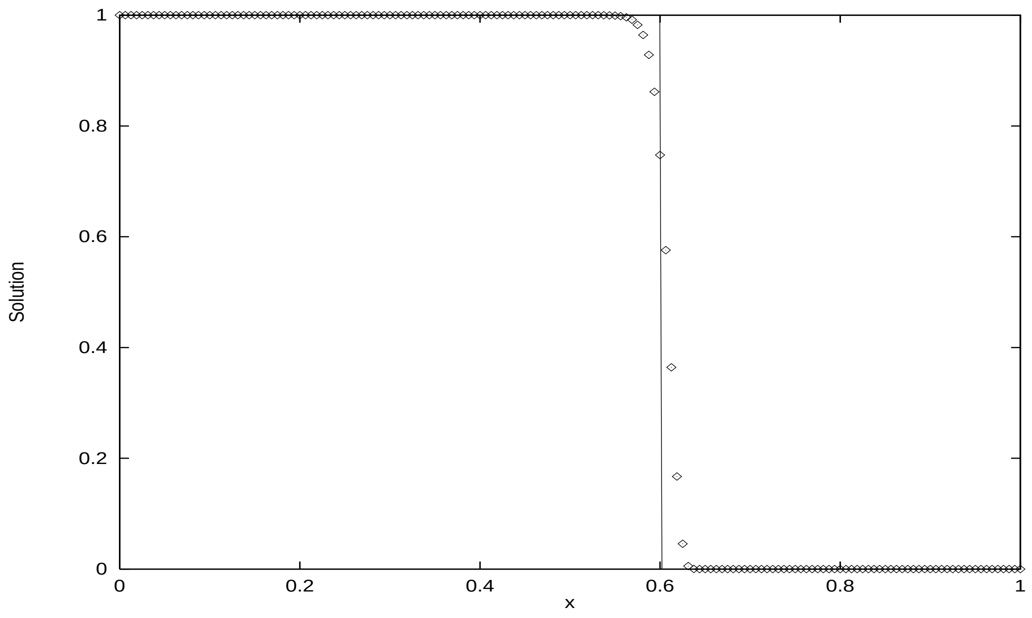


Figure 4.11: The comparison of true (line) and numerical(dot) solutions in 1D Leveque and Yee problem using LEPS control strategy with a uniform mesh of 81 points (top), 161 points(bottom), RTOL=0.01 and ATOL=10^{-5} at time=0.5.

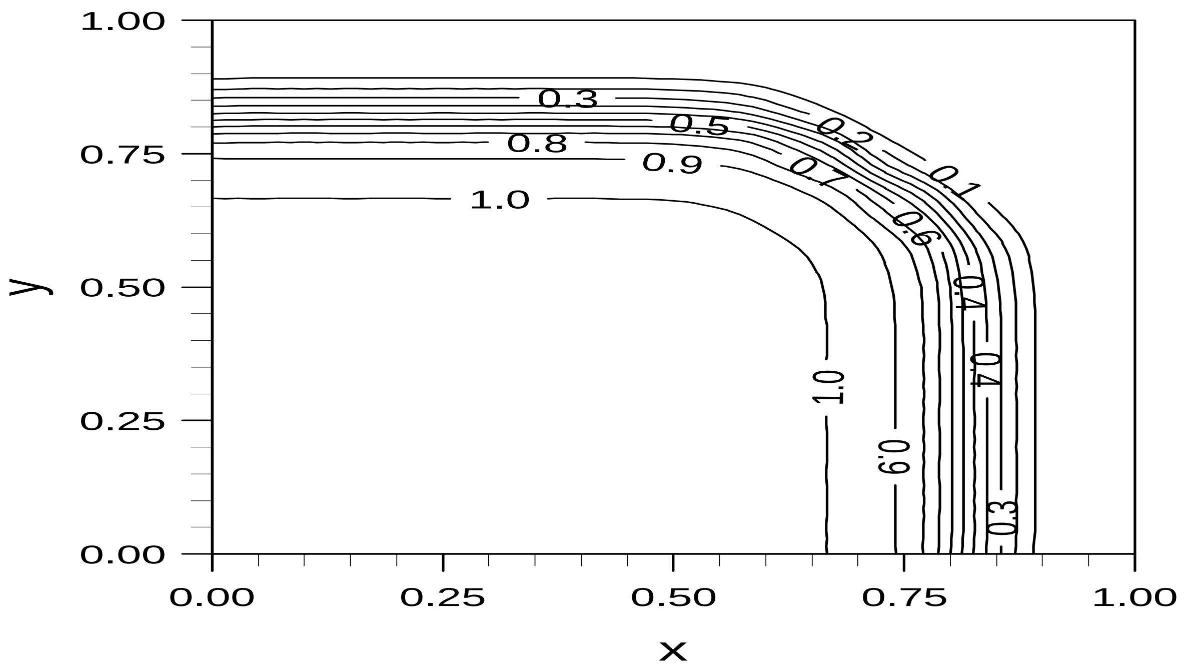


Figure 4.12: Solution of 2D Leveque and Yee problem using the LEPUS strategy on 31 × 31 for balancing factor equal to 0.3 grid at time equal to 0.5.

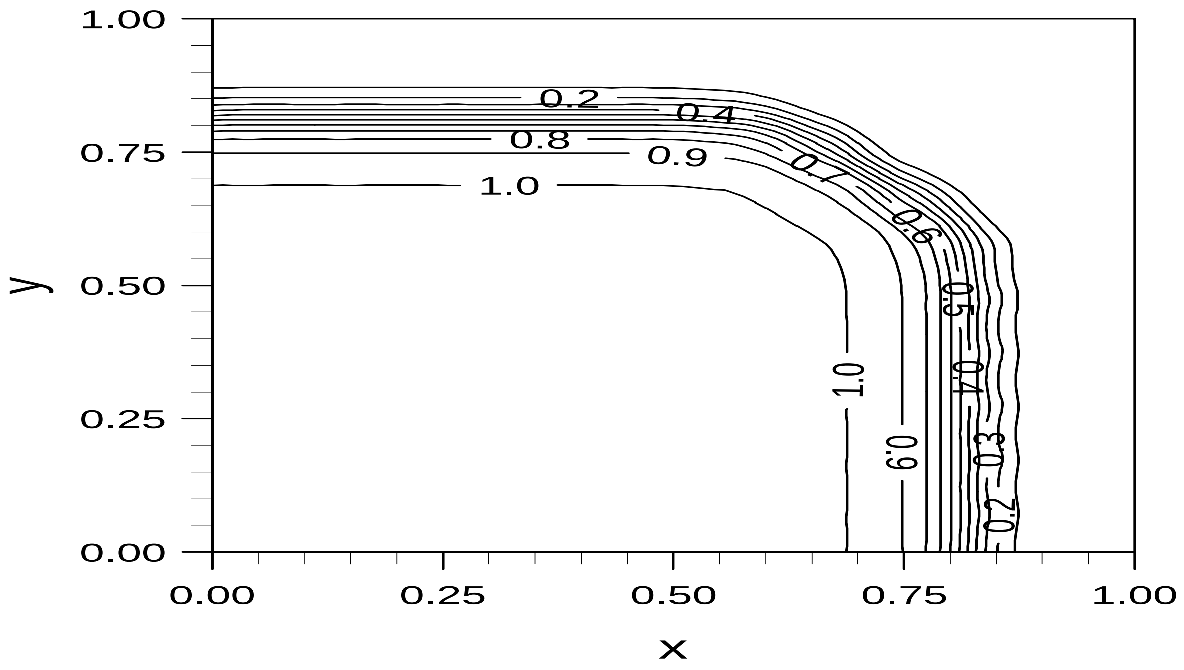


Figure 4.13: Solution of 2D Leveque and Yee problem with the LEPUS control strategy on 41 × 41 for the balancing factor is equal to 0.3 at time=0.6.

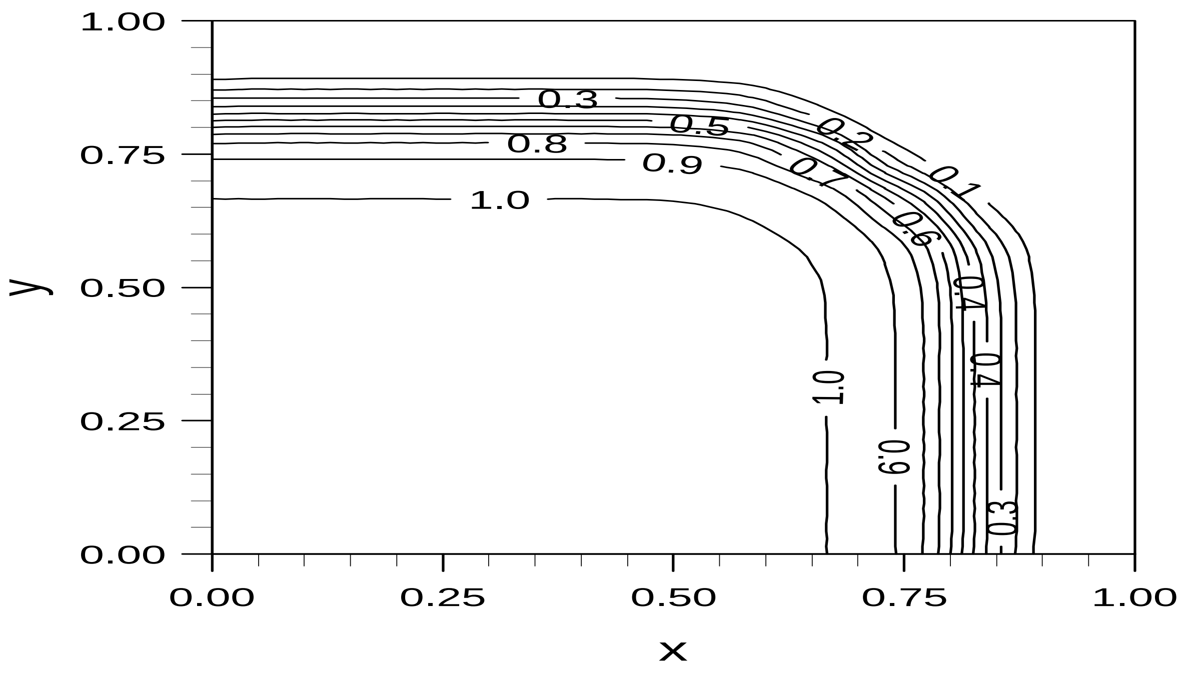


Figure 4.14: Solution of 2D Leveque and Yee problem using the LEPS strategy on 31×31 grid for 0.01 relative tolerance and 1×10^{-4} absolute tolerance.

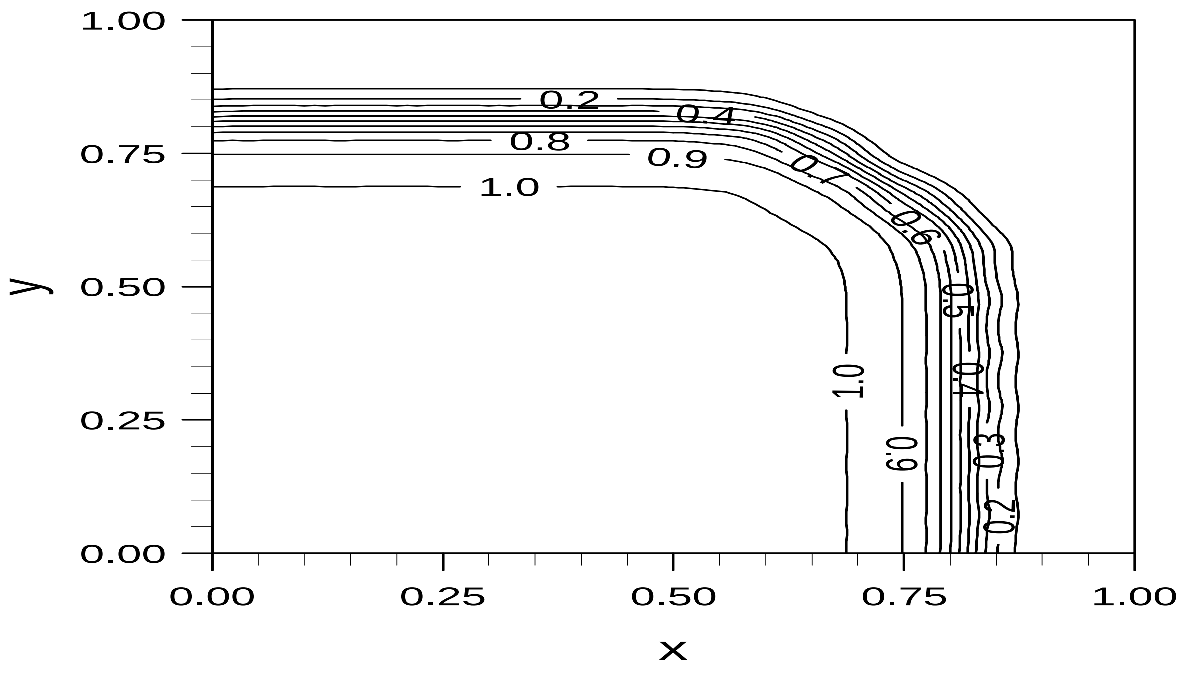


Figure 4.15: Solution of 2D Leveque and Yee problem using the LEPS control strategy on 41×41 grid by using 0.01 relative tolerance and 1×10^{-4} absolute tolerance at time=0.6.

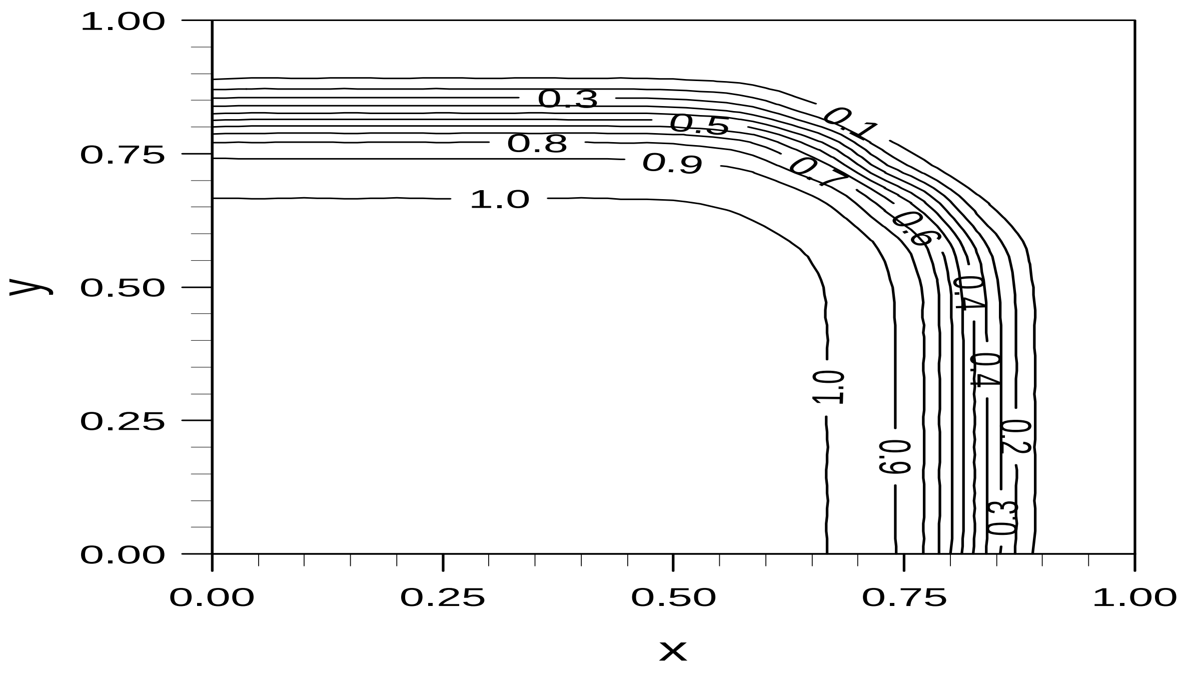


Figure 4.16: Solution of 2D Leveque and Yee problem using the LEPS strategy on 31×31 grid for 0.001 relative tolerance and 1×10^{-4} absolute tolerance.

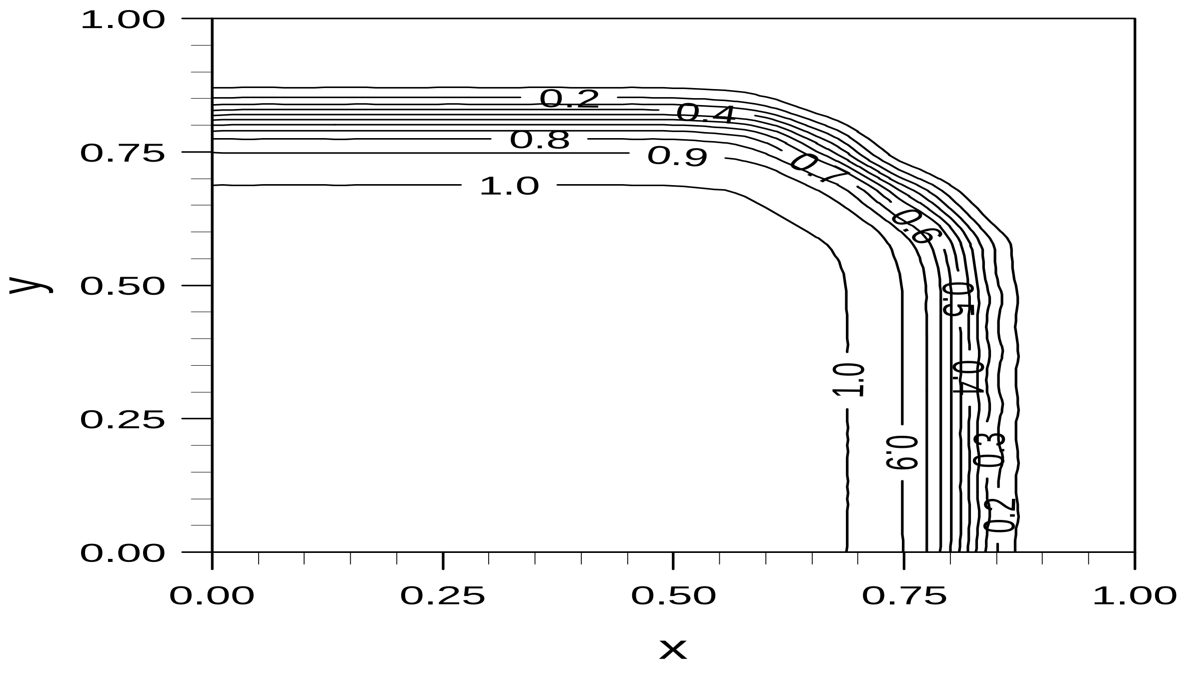


Figure 4.17: Solution of 2D Leveque and Yee problem using the LEPS strategy on 41×41 grid for 0.001 relative tolerance and 1×10^{-4} absolute tolerance at time=0.6.

Chapter 5

Reacting Flow Problems

In this chapter we will discuss the numerical solution of two simple cases of reacting flow problems: an atmospheric dispersion problem and the combustion problem we have been considering. The atmospheric problem is described by the atmospheric diffusion equation given by equation (2.10). In the combustion problem the fluid dynamics and the chemistry are modelled together, while the fluid dynamics governed by the coupled conservation equations:

- one scalar equation for the mass density ρ;
- one vector equation with one, two, or three components for the momentum density ρu;
- and one scalar equation for the energy E.

Solving these equations satisfactorily is one of the important, difficult problems in computational fluid dynamics. A major problem is due to the discontinuity in the solution values which may form a contact discontinuity, shock wave or rare-fraction fan.

Another difficulty posed by reactive flow problems is the spatial resolution needed. The computer time and memory is so demanding for three dimensional calculations, that the chemistry may have to be simplified drastically. Despite this the general trend in reactive flow problems is to use models including an ever large number of reactions in the chemical schemes describing the atmospheric as well as the combustion chemistry. It is desirable, where possible, to minimise the number of chemical species, because much of the computing time is spent in solving the equations describing the chemistry. Some well-known techniques, related to minimise the number of chemical species, such as lumping, sensitivity analysis and quasi-steady-state approximation have discussed in Heard et al. [39]. It this way the number of variables in the scheme reduce significantly.

A critical point in the simulation of reactive flows is the treatment of the source terms which needs special treatment to avoid giving rise to non physical solution values. The reason is that the source term arises from the ODE systems describing the chemical kinetics model in use (see Chapter 3) and thus we expect that the source term will be stiff in time. Thus any numerical method must effectively handle stiff ODEs.

Solving the case in which these chemical kinetics are coupled with fluid dynamics is very different from solving a set of chemical kinetics alone. The large number of computational cells, perhaps ten or hundreds of thousands in a multidimensional domain, usually implies that we cannot afford to store auxiliary information about all the species in each cell between time steps hence some simplifications are needed [58].

The main purpose of the present study is to employ the error balancing approach together with Gauss Seidel iterative method to reactive flow to asses its robustness and accuracy. The reason for adopting this approach is that the error balancing approach gives very promising results on the simulation of the 1D and 2D Leveque and Yee problem [52] (see Chapter 4) and in the work of Berzins [7, 49].

We have used the theta method which is specially designed for the solution of stiff systems with moderate accuracy and automatic control of the local error in time instead of NDF2 method as described in Chapter 3. The numerical experiments have shown that NDF2 method requires the further investigation, in term of, tuning the stepsize strategy and Jacobian evaluation criteria and so the theta method was used.

5.1 An Atmospheric Dispersion Problem

5.1.1 Introduction

The adverse effect of pollutants in daily life has made it vital to study them thoroughly in the context of their effect on the atmosphere. An important example of such pollution includes power station plumes which are concentrated source of NO_x emissions [82]. The photo-chemical reaction of this emitted NO_x and polluted air results in the production of ozone at large distances downwind from the source.

In atmospheric pollution problems achieving high resolution is a difficult challenge because of the large number of species present in the atmosphere. The number of chemical rate equations which need to be solved rises with the number of the species, and for high resolution 3-dimensional calculation, detailed chemical schemes can become prohibitively large. The range of reaction time-scale can lead to stiff systems of differential equations which require more expensive solvers. To avoid this difficulty two strategies are adopted.

The first strategy related with retaining the detailed chemistry for understanding many reactions of pollutants such as NO_x, SO_2 and other organic compounds, and to use 1-D trajectory models or coarse grid models to simulate the reaction/transport problem. Such models are essential in developing an understanding of how chemical species interact to form secondary pollutants. This strategy does not, however, provide the spatial resolution which is needed to understand the complex interaction between multiple source of distributed type.

The second strategy uses a simplified model and a high resolution grid. In this case the problems arise in assessing the role of individual species on the pollutant distribution. Both strategies have advantages and disadvantages and a compromise is necessary between spatial resolution and the number of species. Thus the increasing complexity of practically relevant models requires that new efficient numerical methods should be implemented to solve the underlying extensive systems of time-dependent partial differential equations [82].

5.2 Governing Equations, Spatial Discretization and Controlling Strategy

The complete governing equations of 2D atmospheric problem have been explained in Chapter 2 (see equation (2.10)). The basis of the numerical method is the space discretization of the PDEs discussed in Chapter 2. This approach (known as the "method of lines"), reduces the set of PDEs to a system of ordinary differential equations (ODEs). The system of ODEs can then be solved as an initial value problem, and we have used the theta method [9].

For spatial discretization we have implemented the second order upwind together with van Leer limiter [7] as described in Chapter 2. The novel step here is that instead of controlling the local error per step we have controlled local error per unit step based upon space-time error balancing approach. The motivation was that local per step does not guarantee the best control of the global error, because the relation between the local error and the global error (see for example [7, 49]), is not clear.

5.3 Numerical Results

Problem 1: First we have solved the 1D atmospheric dispersion problem, having 7 species and 7 reactions together with time-dependent photolysis rates as explained in Section (2.5) (see for example [10]). The initial concentrations of the species in molecule/cm^3 have been displayed in Table (3.2).

The polluted air emitted from the source has been assumed to be enriched source of Nitrogen Oxide and has the following concentrations in molecules/cm^3:

$$NO_2 = 1.0 \times 10^{11},$$
$$NO = 1.0 \times 10^{12}.$$

We have assumed a constant wind speed of 0.75 cm/sec in the x-direction. The length of domain is 1 km and the simulation time is two days, i.e., 1.8×10^5 seconds. We have used 21, 41, 81 and 161 points and the mesh size has been kept fixed. The boundary conditions have been described in Problem 1 of Section (4.8)

Problem 2: The 2D atmospheric problem has 20 species and 25 reactions with constant reaction rate from atmospheric chemistry. The initial concentrations in ppm have been given in Table (3.1) and the reaction mechanism has been displayed in appendix (B.0.1).

Similarly to the 1D case we have made the assumption that the polluted air emitted by the source term is enriched with the oxides of Nitrogen having concentration in ppm given as

$$NO_2 = 0.8$$
$$NO = 0.3$$
$$NO_3 = 0.2$$
$$N_2O_5 = 0.3.$$

The domain Ω is $1km \times 1km$, the simulation time is 60 minutes and for convenience we have taken velocities along x-direction and y-direction 1km/min. This problem is highly stiff and the Lipschitz constant is about 1.5×10^7 as the simulation period is 60 minutes which will makes the ODE system stiff (see for example [90]). The boundary conditions have been given in Problem 2 of Section (4.8).

5.3.1 Results Discussion

Two different kinds of error control strategies have been used within the time integration routine

Method A: The LEPS strategy that is controlling local error $\mathbf{le}(t_{n+1})$ (see equation (4.114)) so that

$$\parallel \frac{\mathbf{le}(t_{n+1})}{RTOL \mid \mathbf{V}(t_n) \mid + ATOL} \parallel < 1. \tag{5.1}$$

Method B: The LEPUS strategy that is controlling the local error $\mathbf{le}(t_{n+1})$ (see equation (4.115)) so that

$$\parallel \mathbf{le}(t_{n+1}) \parallel = \epsilon \parallel \hat{\mathbf{est}}(t_{n+1}) \parallel, \tag{5.2}$$

where $\hat{\mathbf{est}}(t_{n+1})$ is the local growth in time of spatial discretisation error estimate and is given by the Section(4.3) and strategy about the selection of the premultiplication factor ϵ has been discussed in the Chapter 4. In these problems we have used $\epsilon = 0.3$ as has been proposed in [49] and experiments show this seems to work well for the problems considered here and Gauss Seidel iterative method has been used to solve the linearised equations.

The following notation has been used to represents the results:

- **NPTS** = The number points used in the spatial discretisation,
- **Nsteps** = The number of integration steps used by the integrator,
- **ATOL** = Absolute error tolerances,
- ϵ = The balancing factor,
- **RTOL** = Relative error tolerance,
- **G-S** = The number of Gauss Seidel iterations.

Methods	NPTS	ATOL or ϵ	RTOL	Nsteps	Fun	Jac	G-S
A	21	10^5	10^{-1}	1340	4376	328	6067
		10^5	10^{-2}	1381	4537	326	6540
		10^5	10^{-3}	1682	5487	293	14086
B	21	0.3		1425	4755	362	11076
A	41	10^5	10^{-1}	1276	4179	304	5881
		10^5	10^{-2}	1332	4652	319	7251
		10^5	10^{-3}	1905	6480	272	16844
B	41	0.3		1486	5042	379	11789
A	81	10^5	10^{-1}	1380	4877	404	7210
		10^5	10^{-2}	1677	6318	448	10465
		10^5	10^{-3}	3068	10250	408	25204
B	81	0.3		1823	6428	489	15282
A	161	10^5	10^{-1}	1290	5133	441	8563
		10^5	10^{-2}	2313	9359	711	15945
		10^5	10^{-2}	5058	16783	571	40668
B	161	0.3		2387	8502	722	19829

Table 5.1: Results of 1D Atmospheric problem with uniform grid using the LEPUS(B) and LEPS(A) control strategies

Point at which Solution given	Methods	ATOL or ϵ	RTOL	Results at t = 180000 seconds	
				NO conc. in mol/cm^3	NO_2 conc. in mol/cm^3
x=0.5	A	10^5	0.1	8.1268×10^{11}	2.8636×10^{11}
			0.01	8.1571×10^{11}	2.8377×10^{11}
			0.001	8.1618×10^{11}	2.8336×10^{11}
	B	0.3		8.1569×10^{11}	2.8380×10^{11}
	B	0.5		8.1520×10^{11}	2.8419×10^{11}

Table 5.2: Table showing the accuracy using the LEPUS(B) and the LEPS(A) strategies for 0.1, 0.01, 0.001 relative tolerances and 1×10^5 absolute tolerance, 161 number of points at time =1.8×10^5 seconds and x=0.5 km.

For the 1D atmospheric problem extensive experiments have been performed in order to check the reliability, efficiency and accuracy of the new proposed scheme, the LEPUS control strategy as given in equation (5.2), against the LEPS (equation (5.1)) control strategy.

In Figure (5.1) we have displayed the results at the origin (where we are solving

Methods	NPTS	ATOL or ϵ	RTOL	Nsteps	Fun	Jac	G-S
A	5×5	1×10^{-5}	1×10^{-1}	863	3829	577	6557
	5×5	1×10^{-5}	1×10^{-2}	1003	4212	695	6613
B	5×5	0.3		826	3549	471	10452
A	10×10	1×10^{-5}	1×10^{-1}	1950	8016	1092	13059
	10×10	1×10^{-5}	1×10^{-2}	1976	8000	1035	12883
B	10×10	0.3		1908	7956	1093	20520
A	15×15	1×10^{-5}	1×10^{-1}	3037	14139	1982	35149
	15×15	1×10^{-5}	1×10^{-2}	3226	14814	2114	31351
	15×15	1×10^{-5}	1×10^{-3}	3401	15295	2304	28571
B	15×15	0.3		3086	13537	1836	36362
A	20×20	1×10^{-5}	1×10^{-1}	4005	17678	22258	43001
	20×20	1×10^{-5}	1×10^{-2}	4276	19101	2601	42125
	20×20	1×10^{-5}	1×10^{-3}	4460	20106	2638	36832
B	20×20	0.3		4151	18095	2342	49363

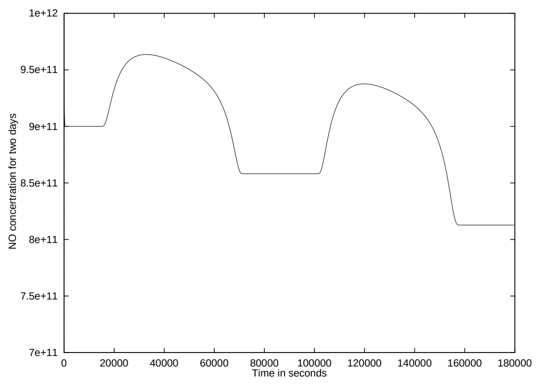
Table 5.3: Results of 2D atmospheric problem with uniform grid using the LEPUS (B) and LEPS (A) control strategies.

The Grid Point at which Solution given	Grid	Methods	ATOL or ϵ	RTOL	Results at t = 60 min	
					NO conc. in *ppm*	NO_2 conc. *in ppm*
The top right corner	15×15	A	10^{-5}	0.1	1.3950×10^{-1}	6.8115×10^{-2}
				0.01	1.3944×10^{-1}	6.8148×10^{-2}
				0.001	1.3944×10^{-1}	6.8196×10^{-2}
		B	0.3		1.3948×10^{-1}	6.8196×10^{-2}
The top right corner	20×20	A	10^{-5}	0.1	1.3790×10^{-1}	6.4963×10^{-2}
				0.01	1.3792×10^{-1}	6.4865×10^{-2}
				0.001	1.379×10^{-1}	6.4910×10^{-2}
		B	0.3		1.3788×10^{-1}	6.4855×10^{-2}

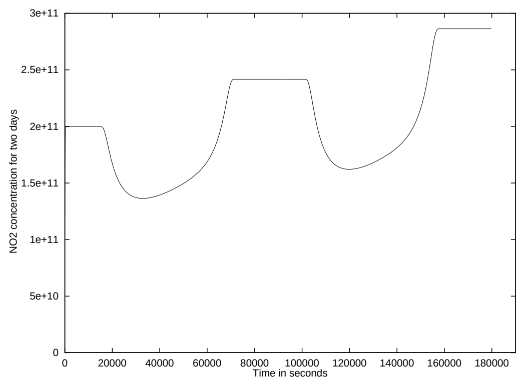
Table 5.4: Table showing the accuracy of the LEPUS(B) and the LEPS(A) control strategy on 15×15 and 20×20 grid respectively at t=60 minutes, with different relative tolerance and fixed absolute tolerance.

a system of ODEs assuming that no advection effect) at time $=1.8 \times 10^5$ seconds for RTOL =0.1, 0.01, by assuming fixed ATOL = 10^5 with LEPS control strategy, while the Figure (5.4) shows the growth of the NO and NO_2 concentrations with the LEPUS control strategy. The comparison shows that there is no difference in the growth of the concentration even with RTOL = 0.01.

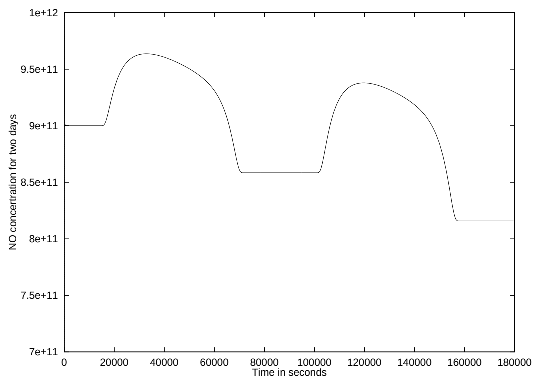
In order to provide more evidence about the accuracy for the LEPUS and different RTOL when the LEPS strategy has been used, we have presented the concentration of these species x=0.5 km in Table (5.2). It is evident that accuracy is improving with decreasing RTOL in LEPS control strategy and it is also clear that using the LEPUS control strategy the results have a superior level of accuracy over RTOL=0.01.



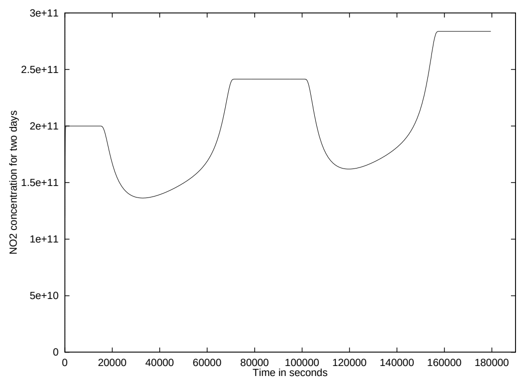
NO concentration with RTOL=0.1



NO_2 concentration with RTOL=0.1



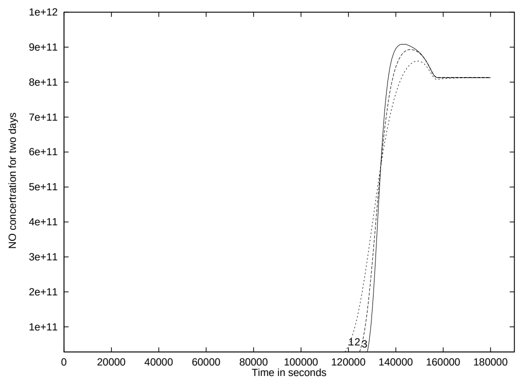
NO concentration with RTOL=0.01



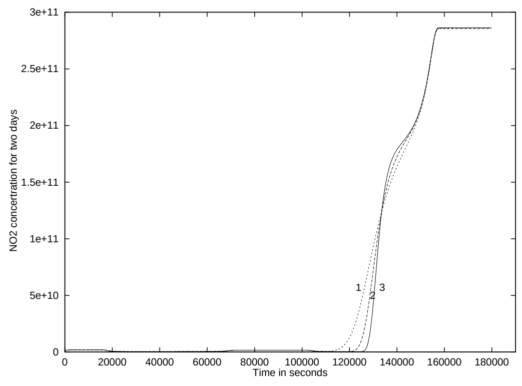
NO_2 concentration with RTOL=0.01

Figure 5.1: Concentration of NO and NO_2 in molecules/cm^3 at origin using LEPS strategy at time =1.8×10^5 seconds for $\triangle x$ =0.00625 km, various relative tolerance and fixed absolute tolerance equal to 1×10^5.

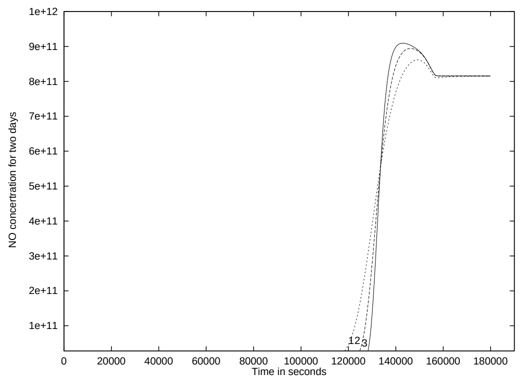
In the second test, we have represented the growth of both species at x=1 km by taking $\triangle x$ = 0.025 km, 0.0125 km, 0.00625 km respectively with both control strategies at time=1.8×10^5 seconds. For the LEPS control strategy we assumed (RTOL = 0.1, 0.01) by taking ATOL = 10^5 and results have been displayed in



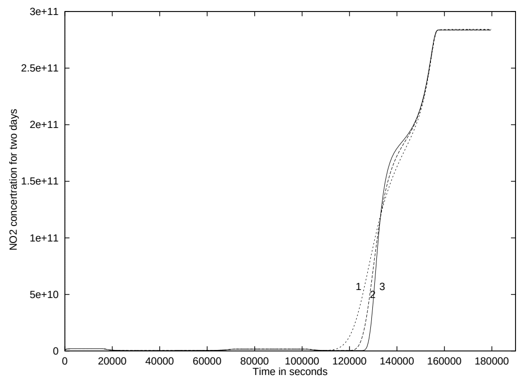
NO concentration with RTOL=0.1



NO_2 concentration with RTOL=0.1



NO concentration with RTOL=0.01

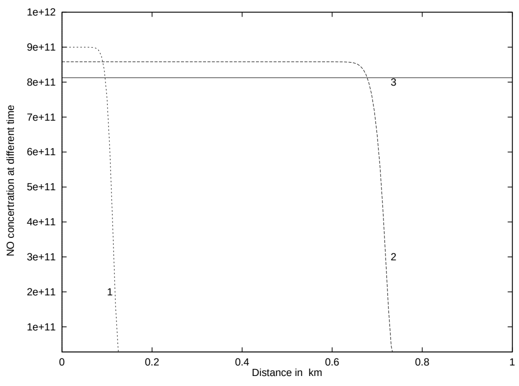


NO_2 concentration with RTOL=0.01

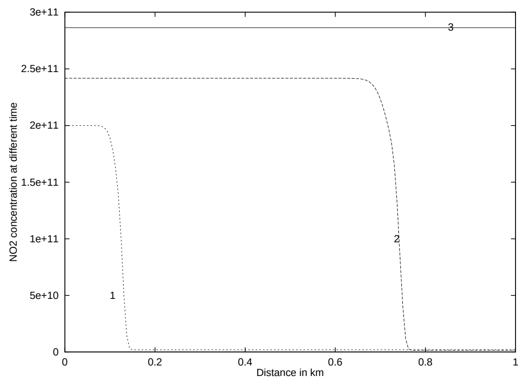
Figure 5.2: Concentration of NO and NO_2 in molecules/cm^3 using LEPS strategy at time $=1.8 \times 10^5$ seconds for $\triangle x$ = 0.025 km(line 1), 0.0125 km(line 2), 0.00625 km(line 3), various relative tolerance and fixed absolute tolerance equal to 1×10^5 at x=1 km.

Figure (5.2), while with the LEPUS control strategy the results have been shown in Figure (5.5), again the comparison shows that there are small differences in results with both control strategies and same trend of accuracy foloows with both strategies as at x=0.5 km.

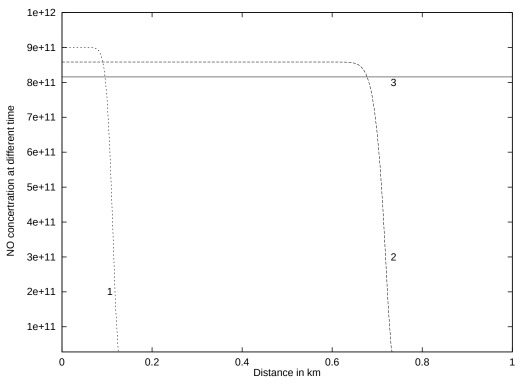
In the third experiment we have shown the growth of above mentioned species at times, t=1.5×10^4 seconds, 9.5×10^4 seconds, 1.8×10^5 seconds with different RTOL (0.1, 0.01) by keeping ATOL fixed with $\triangle x$ = .00625 km. The results are displayed in Figure (5.3). With the same times the experiments have been



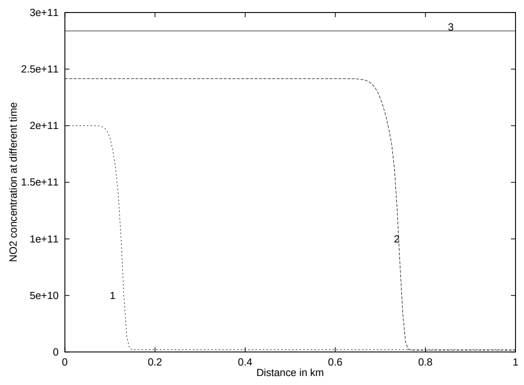
NO concentration with RTOL=0.1



NO_2 concentration with RTOL=0.1



NO concentration with RTOL=0.01

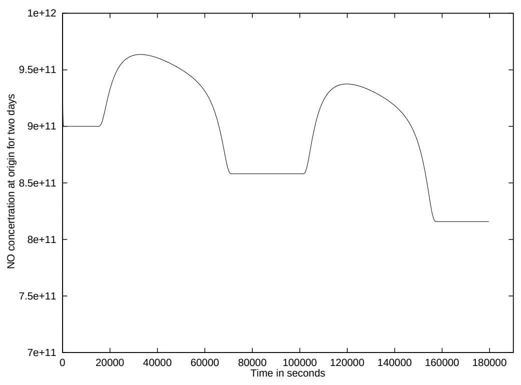


NO_2 concentration with RTOL=0.01

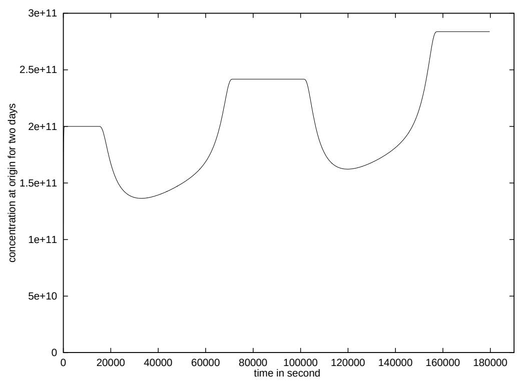
Figure 5.3: Concentration of NO and NO_2 in molecules/cm^3 using LEPS strategy at time =1.5×10^4 seconds (line 1), 9.5×10^4 seconds (line 2), 1.8×10^5 (line 3) seconds for $\triangle x$ =0.00625 km, various relative tolerances and fixed absolute tolerance 1×10^5.

performed with the LEPUS control strategy (see Figure (5.6)), again the graphs look identical and by comparing the computed solution revealed that the accuracy level was same as that x=0.5 km.

For the 2D atmospheric problem we have shown only NO_2 concentration with the the LEPUS (Figure (5.7)) and with LEPS (Figure (5.8)) with RTOL = 0.01. From these Figures it is evident that the results are of comparable accuracy with both strategies. For information about the accuracy of the NO and NO_2 concentration with both strategies, we have presented the results in Table (5.4), and it is evident that with the LEPUS control strategy the computed results have comparable accuracy to that of LEPS control strategy with ATOL=10^{-5} and RTOL=0.001 on both grids.



NO concentration



NO2 concentration

Figure 5.4: Concentration of NO and NO_2 in molecules/cm^3 using the LEPUS strategy at origin at time=1.8×10^5 seconds for $\triangle x = 0.00625$ km and balancing factor (ϵ) equal to 0.3.

The efficiency of both strategies can compared by using the statistics of Tables (5.1) and (5.3), which show the worth of using the local error per unit step control strategy rather than local error per step control strategy in that it delivers the required level of accuracy to let the spatial error dominate.

The numerical experiments on 1D atmospheric problem have revealed that initially the code with local error per unit step strategy is faster than the local error per step control strategy and after some time the reverse situation occurs.

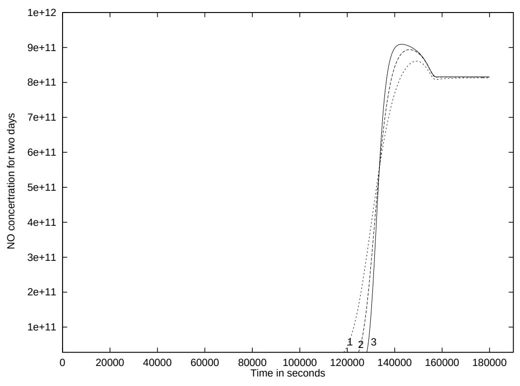
The reason is that initially the spatial error is dominant and with time plumes grow continuously from the left (because of positive velocity) and the code is taking more steps. Once the plume stops growing the local error per unit step control strategy forces the code to do more work in term of the number of integration step as the spatial error is small. In this case it makes sense to have a lower bound on the local growth in time of spatial error (Chapter 4) so that the code does not take too many steps when the local growth in time of spatial error is negligible. This is implemented by passing control to the local error per step based upon the following switch

$$\text{AMAX} \ = \ \max \parallel \mathbf{V}(t_{n+1}) \parallel, \qquad (5.3)$$

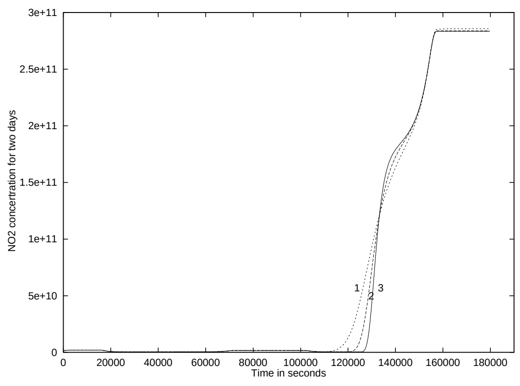
where $\mathbf{V}(t_{n+1})$ is the solution vector at the time t_{n+1} and

$$\text{LSPNRM} \;=\; \max \parallel \hat{\mathbf{est}}(t_{n+1}) \parallel, \tag{5.4}$$

where $\hat{\mathbf{est}}(t_{n+1})$ is the local in time spatial error used as the time tolerance and if LSPNRM $< 10^{-06} \times$ AMAX, then the code is using the local error per step control strategy. These results thus show that the local error per unit step control (equation (5.2)) provides a good balance between efficiency and reliability.

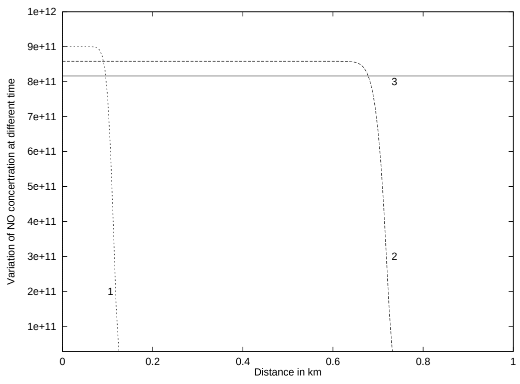


NO concentration at x = 1

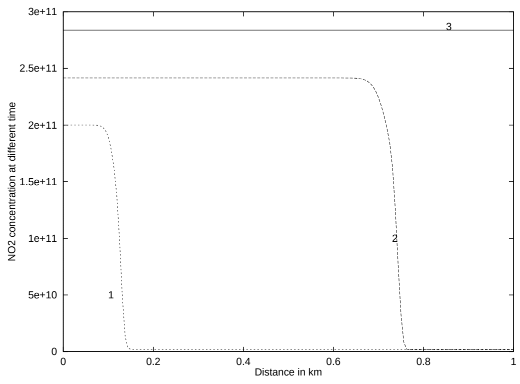


NO_2 concentration at x=1 km

Figure 5.5: Concentration of NO and NO_2 in molecules/cm^3 using the LEPUS strategy at time =1.8×10^5 seconds for $\triangle x = 0.025$ km(line 1), 0.0125 km(line 2), 0.00625 km(line 3) and balancing factor(ϵ) equal to 0.3.



NO concentration



NO_2 concentration

Figure 5.6: Concentration of NO and NO_2 in molecules/cm^3 using LEPUS strategy at time =1.5×10^4 seconds(line 1), 9.5×10^4 seconds (line 2), 1.8×10^5 seconds(line 3) for $\triangle x = 0.00625$ km and balancing factor (ϵ) equal to 0.3.

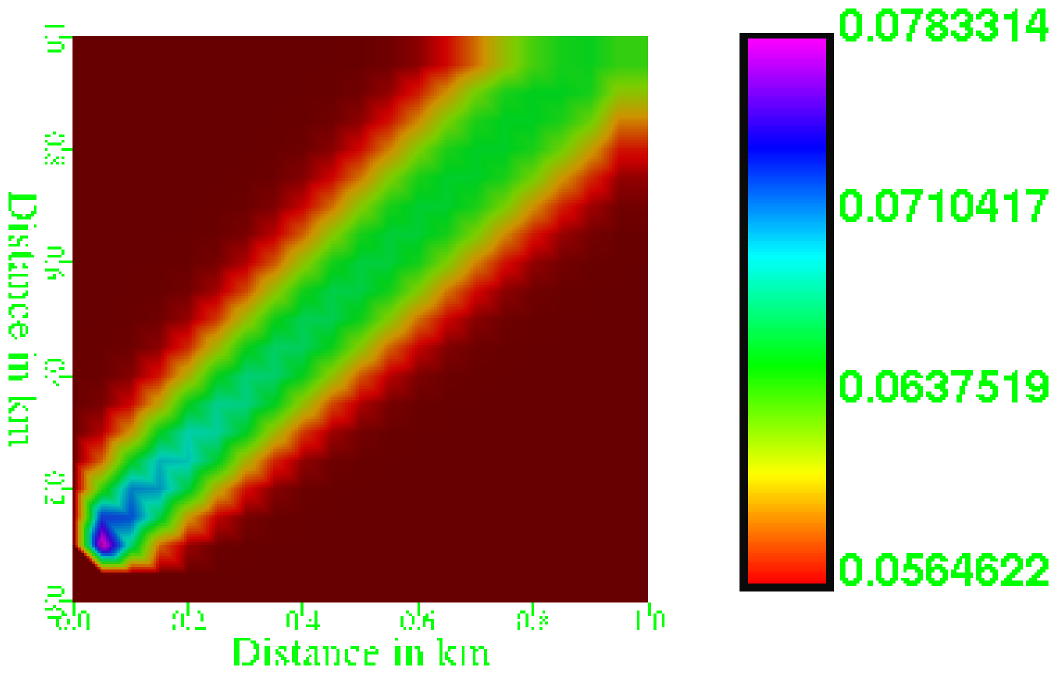


Figure 5.7: Concentration of NO_2 in ppm using LEPUS strategy in 2D atmospheric problem for balancing factor (ϵ) equal to 0.3.

5.4 Combustion Problem

5.4.1 Introduction

Modelling reactive flow in combustion problems is based on a generally accepted set of time-dependent coupled partial differential equations maintaining conservation of density, momentum and energy. Recall that these equations describe the convective motion of the fluid, the chemical reaction among the constituent species and the diffusive transport process such as thermal conduction and molecular diffusion [58]. The Navier Stokes equations are the natural starting point for modelling chemically reacting flow.

There are basically four types of physical processes represented in reactive flow equations. These processes are chemical reactions, diffusive transport, convection and wavelike properties, [58]. The chemical kinetics represents the production and loss of the chemical species, convection describes the motion of fluid quantities in

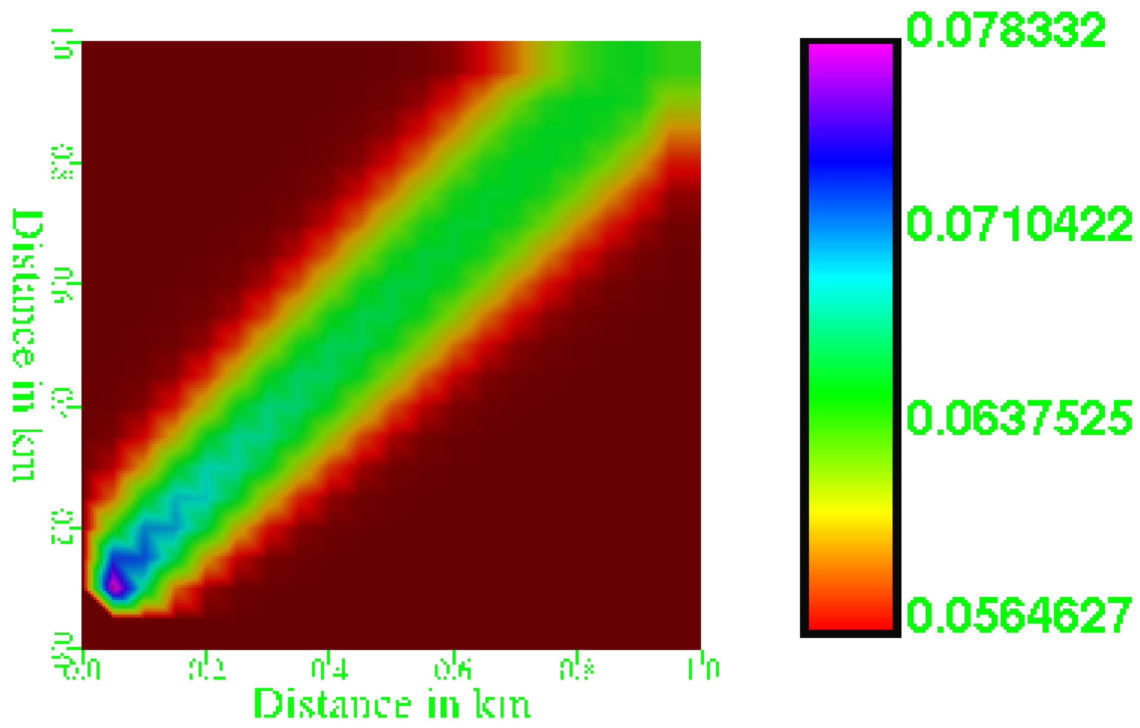


Figure 5.8: Concentration of NO_2 in ppm using LEPS strategy in 2D atmospheric problem for 0.01 relative tolerance and 1×10^{-4} absolute tolerance.

space. The wavelike behaviour are described implicitly in the reactive flow equations by the coupled continuity equations. The important point about wavelike motion is that energy is transferred from one element of the fluid to others by waves that can travel much faster than the fluid velocity.

The main type of wave considered is a shock wave, which moves as a discontinuity through the system. The shock wave heats and compresses the undistributed reactive mixture as it passes through it. The raised temperature triggers chemical reactions, and energy release eventually occurs and the pressure waves are generated, some of which propagate forward and accelerate the shock wave.

The reactions proceed very rapidly after the initiation, which will make the source term stiff [83, 90] in time, hence it is possible that the solution will yield non-physical waves with incorrect speed and incorrect discontinuous in flow properties [28, 83]. Additionally, stiffness problems place restrictions on the time step and the grid spacing, which results in computational inefficiency.

The next task is to solve this complex system of equations. The very complexity of which eliminates the possibility of finding an analytical solution and the only reasonable alternative is to construct the numerical solution. Since the source terms require specialized and possibly costly time integration, the common approach is to use a time splitting to isolate their treatment from the rest of the problem. In this approach, however, a splitting error is introduced. Hence we have used the method of lines approach (see Chapter 2) together with the new splitting approach only at the level of the non-linear equations as described in [12], which avoids any extra splitting error.

To handle the steep spatial fronts, it is natural to apply modern shock-capturing numerical methods for the convective part of the conversation laws. These methods typically require complete analytic expression for the characteristic data, i.e. the eigenvalues and eigenvectors of the linearised convective flux matrix. There are many approaches in common use to handle the steep spatial front and the most popular has been considered in Chapter 1. Due to excellent shock capturing and improved performance of the Marquina method [25] on non-reacting flows, it has also been used for reacting flow, together with theta method as the time integration [9].

5.5 Governing Equations

A brief description of the governing equations of the combustion problem has been given in Chapter 2 and is extended here by using the description given in Fedkiw [28].

The one dimensional compressible flow problem is modelled by the 1D Euler equations and can be written as

$$\mathbf{u}_t + [\mathbf{f}(\mathbf{u})]_x = 0, \tag{5.5}$$

where vector $\mathbf{u}$ and $\mathbf{f}(\mathbf{u})$ are given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{bmatrix}, \tag{5.6}$$

$$E = -p + \frac{\rho u^2}{2} + \rho h, \tag{5.7}$$

and t is the time, x is the spatial dimension, ρ is the density and u is the velocity, E is the energy per unit volume, h is the enthalpy per unit mass, and p is the pressure.

Similarly the two dimensional compressible flow problem is modelled by the 2D Euler equations

$$\mathbf{u}_t + [\mathbf{f}(\mathbf{u})]_x + [\mathbf{g}(\mathbf{u})]_y \;=\; 0, \tag{5.8}$$

where $\mathbf{u}$, $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ has the following form

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix} \quad \mathbf{g}(\mathbf{u}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}, \tag{5.9}$$

where

$$E \;=\; -p + \frac{\rho(u^2 \;+\; v^2)}{2} \;+\; \rho h. \tag{5.10}$$

5.5.1 Energy and Enthalpy

The total energy per unit volume is denoted by E and is equal to the sum of the potential and kinetic energy and is defined by

$$E \;=\; PE + KE \;=\; \rho e + \rho\frac{u^2 + v^2}{2}, \tag{5.11}$$

where e denotes the internal energy per unit mass. The enthalpy per unit mass is defined by

$$h \;=\; e + \frac{p}{\rho}. \tag{5.12}$$

By analogy the enthalpy of the mixture is given by

$$H \;=\; \frac{E+p}{\rho}. \tag{5.13}$$

Using equation (5.12) we can write equation (5.11) as follows

$$E \;=\; -p + \rho\frac{u^2 + v^2}{2} + \rho h, \tag{5.14}$$

and similarly the total enthalpy is given by

$$H \;=\; h + \frac{u^2 + v^2}{2}. \tag{5.15}$$

A “perfect”gas is defined as a gas in which intermolecular forces are neglected and in a perfect gas the internal energy, enthalpy and specific heat are a function of the temperature only. So we have that, for perfect gas,

$$h = h(T), \qquad e = e(t), \tag{5.16}$$

$$c_p = c_p(T), \qquad c_v = c_v(t), \tag{5.17}$$

and c_p is the specific heat at constant pressure. For a perfect gas we have two further relations [28]

$$dh(T) = c_p(T)dT, \qquad de(T) = c_v(T)dT. \tag{5.18}$$

Integration of both sides of the above equation gives

$$\int_{h(0)}^{h(T)} ds = \int_0^T c_p(s)ds, \tag{5.19}$$

and the further simplification yields the following form of the above equation

$$h(T) - h(0) = \int_0^T c_p(s)ds, \tag{5.20}$$

$$h(T) = h(0) + \int_0^T c_p(s)ds, \tag{5.21}$$

where s is the dummy variable of integration. A perfect gas can further be divided into a thermally perfect gas (specific heats are functions of the temperature) and calorically perfect gas(specific heats are constant). For the calorically perfect gas the equation (5.21) has the form

$$h(T) = h^f + c_pT, \tag{5.22}$$

where $h^f = h(0)$ is the enthalpy per unit mass at 0K, and also called the heat of formulation and is constant can be found in the JANAF thermodynamics Table [77]. For thermally perfect gas the equation (5.21) can be written as

$$h(T) = h^f + \int_0^T c_p(s)ds. \tag{5.23}$$

5.5.2 Equation of State

The equation of the state for the perfect gas is, [56],

$$p = \rho\tilde{R}T, \tag{5.24}$$

and in the above equation $\tilde{R}$ is the specific gas constant and given by the following relation

$$\tilde{R} = \frac{R_u}{W}, \tag{5.25}$$

and $R_u = 8314$ Joules per kilomole degree Kelvin is the universal gas constant, and W is the molecular weight of the gas. W can be found in the JANAF Thermochemical Tables [77].

5.5.3 Gamma

The ratio of the specific heats is denoted by γ and given by [56]

$$\gamma = \frac{c_p}{c_v}. \tag{5.26}$$

Another useful relation is given by

$$c_p - c_v = \tilde{R}, \tag{5.27}$$

which is valid for both a calorically perfect and a thermally perfect gas. With the help of the equations (5.26) and (5.27) the above equation can be written as

$$\gamma = \frac{c_p}{c_p - \tilde{R}}, \tag{5.28}$$

which is valid for both calorically perfect and thermally perfect gases.

5.6 Multiple Species

The 2D Euler equations can be modified in such a way that the flow of more than one species can be considered. The modified form of 2D Euler Equations for multi-species flows are (see for example Ton et al. [83]):

$$\mathbf{u}_t + [\mathbf{f}(\mathbf{u})]_x + [\mathbf{g}(\mathbf{u})]_y = 0, \tag{5.29}$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho Y_1 \\ \vdots \\ \rho Y_{NS-1} \end{bmatrix} \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \\ \rho u Y_1 \\ \vdots \\ \rho u Y_{NS-1} \end{bmatrix} \quad \mathbf{g}(\mathbf{u}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \\ \rho v Y_1 \\ \vdots \\ \rho v Y_{NS-1} \end{bmatrix}, \tag{5.30}$$

and where

$$E \;=\; -p + \frac{\rho(u^2 \;+\; v^2)}{2} \;+\; \rho h, \tag{5.31}$$

NS represents the number of species and Y_i is the mass fraction of species i and $Y_{NS} \;=\; 1 - \sum_{i=1}^{NS-1} Y_i$.

5.6.1 Energy and Enthalpy

By analogy with equation (5.12) we define the enthalpy, h for the mixture of gas as follows [28]

$$h \;=\; e \;+\; \frac{p}{\rho} \;=\; \sum_{i=1}^{NS} Y_i e_i \;+\; \frac{\sum_{i=1}^{NS} p_i}{\rho} \;=\; \sum_{i=1}^{NS} Y_i\left(e_i \;+\; \frac{p_i}{\rho Y_i}\right) = \sum_{i}^{NS} Y_i h_i, \tag{5.32}$$

where e_i, p_i and h_i represents the internal energy, partial pressure and enthalpy per unit mass of the ith gas respectively. Similarly to equation (5.21) the enthalpy for a perfect gas can be written as

$$h_i(T) \;=\; h_i^f \;+\; \int_0^T c_{p,i}(s)ds, \tag{5.33}$$

where h^f is the heat of formation and can be found from the JANAF tables of thermodynamics [77] and like that of equation(5.22) equation (5.33) can be written for the ith calorically perfect gas as:

$$h_i(T) \;=\; h_i^f \;+\; c_{p,i}T. \tag{5.34}$$

Now the combination of equation(5.32) and (5.33) gives

$$h \;=\; \sum_{i=1}^{NS} Y_i h_i^f + \int_0^T \sum_{i=1}^{NS} Y_i c_{p,i}(s)ds \;=\; \sum_{i=1}^{NS} Y_i h_i^f + \int_0^T c_p(s)ds, \tag{5.35}$$

where c_p is the total specific heat at constant pressure of the mixture, and for the mixture of calorically perfect gas this can be written as

$$h = \sum_{i=1}^{NS} Y_i h_i^f + c_p T. \tag{5.36}$$

5.6.2 Equation of State

When multi-species flow is considered with the assumption that each species is a thermally perfect gas, then the partial pressure of each species is given as, [28],

$$p_i = \rho Y_i R_i T, \tag{5.37}$$

where R_i is the specific gas constant for each species and is given as

$$R_i = \frac{R_u}{W_i}, \tag{5.38}$$

and where W_i represents the molecular weight of the ith species and R_u is the universal gas constant. Now R for the multi-species flow is defined as

$$R = \sum_{i=1}^{NS} Y_i R_i, \tag{5.39}$$

and then the equation of state for multi-species flow has the following form

$$p = \sum_{i=1}^{NS} p_i = \sum_{i=1}^{NS} \rho Y_i R_i T = \rho \left(\sum_{i=1}^{NS} Y_i R_i \right) = \rho R T, \tag{5.40}$$

which is valid for both calorically perfect and thermally perfect gases, [28].

5.6.3 Gamma

Similarly to equation (5.39), the specific heat c_p and W can be defined for the gas mixture as follows:

$$c_p = \sum_{i=1}^{NS} Y_i c_{p,i}, \tag{5.41}$$

and

$$W = \frac{1}{\sum_{i=1}^{NS} \frac{Y_i}{W_i}}, \tag{5.42}$$

then γ for the mixture of gases is defined as

$$\gamma = \frac{c_p}{c_p - \frac{R_u}{W}}. \tag{5.43}$$

and c_p and W are defined by equations (5.41) and (5.42) respectively (for example see [28]). Recall that for the mixture of the calorically perfect gases, $c_{p,i}$ is constant for each i, which implies that $c_p = c_p(Y_i)$ is a function of the mass fraction only. As $W = W(Y_i)$ is also a function of the mass fraction, it is evident that $\gamma = \gamma(Y_i)$ is function of the mass fraction.

On the other hand for a thermally perfect gas, each $c_{p,i} = c_{p,i}(T)$ is a function of temperature and it implies that $c_p = c_p(Y_i, T)$. Hence $\gamma = \gamma(Y_i, T)$ is function of both the mass fraction and the temperature (see for example [28]).

5.6.4 Mean Molecular Weight for a Mixture

The equation (5.42) is different then equations (5.39) and (5.41). The difference is that both R and c_p are defined per unit mass, while W is defined per mole. Thus when defining W for the mixture, we use mole fractions, not mass fractions. So the mean molecular weight for a mixture is defined by (see for example [28])

$$W = \sum_{i=1}^{NS} X_i W_i, \tag{5.44}$$

where X_i is the mole fraction of species i. This has been defined along the lines of equations (5.39) and (5.41).

Now,

$$X_i = \frac{\tilde{m}_i}{\tilde{m}} = \frac{\frac{\tilde{M}_i}{W_i}}{\frac{\tilde{M}}{W}} = \frac{\tilde{M}_i}{\tilde{M}}\frac{W}{W_i} = Y_i\frac{W}{W_i}, \tag{5.45}$$

where $\tilde{m}_i$ is the moles of species i, $\tilde{m}$ is the total moles of the mixture, $\tilde{M}_i$ is the mass of species i, $\tilde{M}$ is the total mass of the mixture, W_i is the atomic weight of species i and W is the atomic weight of the mixture.

If we sum both sides of above equation, we have that

$$\sum_{i=1}^{NS} X_i = \sum_{i=1}^{NS} Y_i\frac{W}{W_i}. \tag{5.46}$$

The left hand side of the above equation represents the sum of the mole fraction of all species and should be equal to 1. Then we have that

$$1 = W\sum_{i=1}^{NS}\frac{Y_i}{W_i}, \tag{5.47}$$

which gives that

$$W = \frac{1}{\sum_{i=1}^{NS}\frac{Y_i}{W_i}}, \tag{5.48}$$

which gives the equation (5.42).

5.7 Chemical Reactions

The 2D equations can be modified for reactive flow problems by incorporating the chemical reactions. These equations represent the convective motion of the fluid and chemical reactions among the constituent species. Hence for multiple species flow with chemical reactions the 2D Euler Equations are given by (see [28, 83])

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u}) + \mathbf{g}(\mathbf{u}) = \psi(\mathbf{u}) \tag{5.49}$$

where **u**, **f**(**u**) and **g**(**u**) are given by equation (5.6) and the source terms ψ are given by

$$\psi(\mathbf{u}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dot{w}_1(T, p, Y_1, Y_2, \cdots, Y_{NS-1}) \\ \vdots \\ \vdots \\ \dot{w}_{NS-1}(T, p, Y_1, Y_2, \cdots, Y_{NS-1}) \end{bmatrix},$$

and $\dot{w}_i$ represents the mass production rate of the ith species and has been discussed in Chapter 2 (also see [28] for more details).

5.8 Temperature, Gamma, Specific Heat, and Enthalpy

The factor γ can be calculated from the conserved variable and the specific heat c_p, which implies that the next task is to calculate c_p (see equation (5.41)), which means that we need to calculate $c_{p,i}$ for each species. These can be evaluated with the help of polynomial fit as described in Kee et al.[45] if the temperature is known. Hence the major task is to calculate temperature from conserved variables. The most general form of the energy equation is given by

$$E \;=\; -p + \rho \frac{u^2 + v^2}{2} + \rho h, \tag{5.50}$$

and the equation of state for the mixture of gas is given by

$$p \;=\; \rho \left[\sum_{i=1}^{NS} Y_i R_i \right] T. \tag{5.51}$$

The combination of equation (5.50) and equation(5.51) means that the temperature can be written as

$$\begin{aligned} T \;&=\; \frac{-E + \frac{\rho(u^2+v^2)}{2} + \rho \left(\sum_{i=1}^{NS} Y_i h_i^f + \int_0^T c_p(s) \right)}{\left(\rho(\sum_{i=1}^{NS} Y_i R_i \right)}, \\ &=\; C_3 \int_0^T c_p(s) ds + C_4, \end{aligned} \tag{5.52}$$

where C_3 and C_4 are constants if the conserved variables are fixed. With the assumption of calorically perfect gas the equation (5.52) can be written as

$$T = \frac{C_4}{1 - C_3 c_p}. \tag{5.53}$$

However for the thermally perfect gas the equation (5.52) is implicit in temperature, and for simplicity we write it in the following form

$$\begin{aligned} T &= \frac{-E + \frac{\rho(u^2+v^2)}{2} + \rho h(T)}{\rho\left(\sum_{i=1}^{NS} Y_i R_i\right)}, \\ &= C_1 + C_2 h(T), \end{aligned} \tag{5.54}$$

where C_1 and C_2 are constants if the conserved variables are fixed and the above equation has the form

$$f(T) = T - C_1 - C_2 h(T). \tag{5.55}$$

In the above equation $\frac{dh(T)}{dt} = c_p(T)$ (see equation (5.18)), and $C_2 = \frac{1}{R}$ (equation (5.39)) and it is also given in the previous Section that $\gamma = \frac{c_p}{c_p - R}$, then the combination of all these quantities enables that the equation (5.55) to be written as

$$\frac{df(T)}{dT} = 1 - C_2 \frac{dh(T)}{dT} = 1 - C_2 c_p(T) = 1 - \frac{c_p(T)}{R} = \frac{-1}{\gamma - 1}, \tag{5.56}$$

where γ is the function of temperature and which is always greater that one, so we have that

$$\frac{df(T)}{dT} = \frac{-1}{\gamma - 1} < 1, \tag{5.57}$$

and which guarantees that the function is strictly decreasing function. So the equation (5.55) can be solved by using a Newton Raphson iteration [16]. Hence with the implementation of the Newton Raphson iteration the equation (5.55) can be cast in the following form.

$$T_{n+1} = T_n - f(T_n)\left[\frac{T_n - T_{n-1}}{f(T_n) - f(T_{n-1})}\right]. \tag{5.58}$$

For performing above iteration there is a need to evaluate the enthalpy $h(T)$ as the function of temperature. For this purpose the integration of the equation (5.18) with the reference temperature 298 Kelvin, gives

$$\int_{h_i(298)}^{h_i(T)} ds = \int_{298}^{T} c_{p,i}(s) ds, \tag{5.59}$$

$$h_i(T) \;=\; h_i^{298} \;+\; \int_{298}^{T} c_{p,i}(s)ds, \tag{5.60}$$

where s and $h_i^f = h_i^{298}$ are the dummy variable of integration and enthalpy per unit mass at 298K for species i and also called the heat of formation at 298K, and is given in [77]. With the assumption of the calorically perfect gas, then 298K can used to evaluate the constant value of $c_{p,i}$. Then with notation of $c_{p,i}^{298}$, the equation (5.60) can be written as

$$h_i(T) \;=\; h_i^{298} \;+\; c_{p,i}^{298}(T \;-\; 298), \tag{5.61}$$

for a calorically perfect gas with reference temperature 298K (see for example Fedkiw [28]). Now the enthalpy as a function of temperature is given by equation (5.32). To speed up the code the similar procedure as described in [28] has been adopted , i.e. at the beginning of the code, the table of $h_i(T)$'s for each species and every integer number between 300K and 4800K have been created by the polynomial fit (see for example Kee et al.[45]), and during the computation, for the non-integral value of the temperature, $h_i(T)$'s have been obtained by interpolation.

5.9 Eigenvalues and Eigenvectors

Upwind scheme are widely used for the simulation of the reacting flow problems due to their excellent shock capturing ability, but may require a complete analysis of the Jacobian matrix of the PDEs system in term of the eigenvalues and eigenvectors. On the other hand when attempting to simulate complex phenomenon, a system of equations will almost certainly have very complicated convection terms. It is hard to find the Jacobian matrix of a convection term with respect to the conserved variables.

If the eigensystem is slightly perturbed then the characteristic fields are changed and in some cases upwind for one field is downwind for another [28]. In the following Section we will write the eigenvalues and eigenvectors, the detail is given in [28]

5.9.1 1D Euler Equation

Consider the equations (5.5), (5.6) and (5.7). Recall that for the thermally perfect gas, the equation (5.21) gives the enthalpy per unit mass as

$$h \;=\; h^f \;+\; \int_{0}^{T} c_p(s)ds, \tag{5.62}$$

which is the function of temperature, and for any conserved variable, the equation (5.62) can be written as

$$\frac{dh}{dV_c} = c_p(T)\frac{dT}{dV_c}, \tag{5.63}$$

which express the derivatives of h with respect to conserved variables. For the calorically perfect gas the enthalpy per unit mass, see equation (5.22) is given by

$$h(T) = h^f + c_pT, \tag{5.64}$$

and c_p is constant. Then the derivative of h can be expressed in the following way

$$\frac{dh}{dV_c} = c_p(T)\frac{dT}{dV_c}. \tag{5.65}$$

Now the equations (5.63) and (5.65) are identical except the fact that c_p is the function of temperature in equation (5.63). The equation (5.7) can be written for pressure as follow

$$p = -E + \frac{\rho u^2}{2} + \rho h, \tag{5.66}$$

and the derivative with respect to the conserved variables are given as

$$\frac{dp}{d\rho} = -\frac{u^2}{2} + h + \rho c_p\frac{dT}{d\rho}, \tag{5.67}$$

$$\frac{dp}{(d\rho u)} = u + \rho c_p\frac{dT}{(d\rho u)}, \tag{5.68}$$

$$\frac{dp}{dE} = -1 + \rho c_p\frac{dT}{dE}, \tag{5.69}$$

with the help of equations (5.63) and (5.65). Now the derivative of the equation (5.24) with respect to conserved variables gives that

$$\frac{dp}{d\rho} = \tilde{R}T + \rho\tilde{R}\frac{dT}{d\rho}, \tag{5.70}$$

$$\frac{dp}{d(\rho u)} = \rho\tilde{R}\frac{dT}{d(\rho u)}, \tag{5.71}$$

$$\frac{dp}{dE} = \rho\tilde{R}\frac{dT}{dE}, \tag{5.72}$$

which can used to eliminated the derivative of T in equations(5.67-5.69). We can then solve for the derivative of p to obtain

$$\frac{dp}{d\rho} = (\gamma - 1)(\frac{u^2}{2} - h + c_pT), \tag{5.73}$$

$$\frac{dp}{d(\rho u)} = (\gamma - 1)(-u), \tag{5.74}$$

$$\frac{dp}{dE} = (\gamma - 1), \tag{5.75}$$

which will be needed while calculating the Jacobian matrix. Now the Jacobian matrix of $\mathbf{f}(\mathbf{u})$ is given as

$$uI + \left(\frac{dp}{d\rho}\mathbf{J}_f \quad \frac{dp}{d(\rho u)}\mathbf{J}_f \quad \frac{dp}{dE}\mathbf{J}_f\right) + (1\mathbf{J}_b \quad u\mathbf{J}_\mathbf{b} \quad H\mathbf{J})_b)^T \tag{5.76}$$

and the notation M^T stands for the transpose of the matrix M, I is the 3×3 identity matrix and

$$\mathbf{J}_f = \begin{bmatrix} 0 \\ 1 \\ u \end{bmatrix} \quad \mathbf{J}_b = \begin{bmatrix} -u \\ 1 \\ 0 \end{bmatrix}. \tag{5.77}$$

The eigenvalues of above mentioned Jacobian matrix are

$$\lambda_1 = u - \tilde{c}, \quad \lambda_2 = u, \quad \lambda_3 + u + \tilde{c}, \tag{5.78}$$

and corresponding the left eigenvectors $\mathbf{L}^{(1)}$, $\mathbf{L}^{(2)}$, $\mathbf{L}^{(3)}$ and the right eigenvectors $\mathbf{R}^{(1)}$, $\mathbf{R}^{(2)}$, $\mathbf{R}^{(3)}$ are

$$\mathbf{L}^{(1)} = \left(\frac{b_2}{2} + \frac{u}{2\tilde{c}}, \quad \frac{-b_1 u}{2} - \frac{1}{2\tilde{c}}, \quad \frac{b_1}{2}\right), \tag{5.79}$$

$$\mathbf{L}^{(2)} = (1 - b_2, \quad b_1 u, \quad -b_1), \tag{5.80}$$

$$\mathbf{L}^{(3)} = \left(\frac{b_2}{2} - \frac{u}{2\tilde{c}}, \quad \frac{-b_1 u}{2} + \frac{1}{2\tilde{c}}, \quad \frac{b_1}{2}\right), \tag{5.81}$$

$$\mathbf{R}^{(1)} = \begin{bmatrix} 1 \\ u - \tilde{c} \\ H - u\tilde{c} \end{bmatrix}, \quad \mathbf{R}^{(2)} = \begin{bmatrix} 1 \\ u \\ H - \frac{1}{b_1} \end{bmatrix}, \quad \mathbf{R}^{(3)} = \begin{bmatrix} 1 \\ u + \tilde{c} \\ H + u\tilde{c} \end{bmatrix}, \tag{5.82}$$

where

$$\tilde{c} = \sqrt{\frac{\gamma p}{\rho}}, \tag{5.83}$$

and

$$b_1 = \frac{\gamma - 1}{\tilde{c}^2} \qquad b_2 = 1 + b_1 u^2 - b_1 H. \tag{5.84}$$

5.9.2 2D Euler

Consider the equation (5.8-5.10). In order to obtain the eigenvalues and eigenvectors of the Jacobian matrix $\mathbf{f}(\mathbf{u})$ set $\tilde{A} = 1$ and $\tilde{B} = 0$, and similarly the eigenvalues and eigenvectors of $\mathbf{g}(\mathbf{u})$) can be obtained by setting $\tilde{A} = 0$ and $\tilde{B} = 1$ in the equations given below (see for detail Fedkiw [28]).

From equation (5.10) the pressure is given by

$$p = -E + \frac{\rho(u^2 + v^2)}{2} + \rho h, \tag{5.85}$$

then the derivatives with respect to the conserved variables are

$$\frac{dp}{d\rho} = \frac{-(u^2 + v^2)}{2} + h + \rho c_p \frac{dT}{d\rho}, \tag{5.86}$$

$$\frac{dp}{d(\rho u)} = u + \rho c_p \frac{dT}{d(\rho u)}, \tag{5.87}$$

$$\frac{dp}{d(\rho v)} = v + \rho c_p \frac{dT}{d(\rho v)}, \tag{5.88}$$

$$\frac{dp}{dE} = -1 + \rho c_p \frac{dT}{dE}. \tag{5.89}$$

Then the derivative of equation (5.24) with respect to the conserved variables are given as

$$\frac{dp}{d\rho} = \tilde{R}T + \rho\tilde{R}\frac{dT}{d\rho}, \tag{5.90}$$

$$\frac{dp}{d(\rho u)} = \rho\tilde{R}\frac{dT}{d(\rho u)}, \tag{5.91}$$

$$\frac{dp}{d(\rho v)} = \rho\tilde{R}\frac{dT}{d(\rho v)}, \tag{5.92}$$

$$\frac{dp}{dE} = \rho\tilde{R}\frac{dT}{dE}. \tag{5.93}$$

The above derivatives can be used to eliminate the derivative of T in equations (5.86-5.89) and finally we have that

$$\frac{dp}{d\rho} = (\gamma - 1)(\frac{u^2 + v^2}{2} - h + c_p T), \tag{5.94}$$

$$\frac{dp}{(d\rho u)} = (\gamma - 1)(-u), \tag{5.95}$$

$$\frac{dp}{(d\rho v)} = (\gamma - 1)(-v), \tag{5.96}$$

$$\frac{dp}{dE} = (\gamma - 1), \tag{5.97}$$

which will be utilized while calculating the Jacobian of both $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$. Now the Jacobian matrix is given by

$$\hat{u}I + \left(\frac{dp}{d\rho}\mathbf{J}_f \ \ \frac{dp}{d(\rho u}\mathbf{J}_f \ \ \frac{dp}{d(\rho v}\mathbf{J}_f \ \ \frac{dp}{dE}\mathbf{J}_f\right) + (1\mathbf{J}_b \ \ u\mathbf{J}_b \ \ v\mathbf{J}_b \ \ H\mathbf{J}_b)^T. \tag{5.98}$$

In the above equation M^T means the transpose of the matrix M, I is the 4×4 identity matrix, and

$$\mathbf{J}_f = \begin{bmatrix} 0 \\ \tilde{A} \\ \tilde{B} \\ \hat{u} \end{bmatrix} \quad \mathbf{J}_b = \begin{bmatrix} -\hat{u} \\ \tilde{A} \\ \tilde{B} \\ 0 \end{bmatrix}. \tag{5.99}$$

Next list of the eigenvalues, the left and right eigenvectors of the Jacobian matrix are given by

$$\lambda_1 = \hat{u} - \tilde{c}, \quad \lambda_2 = \lambda_3 = \hat{u}, \quad \lambda_4 = \hat{u} + \tilde{c}, \tag{5.100}$$

$$\mathbf{L}^{(1)} = \left(\frac{b_2}{2} + \frac{\hat{u}}{2\tilde{c}}, \; -\frac{b_1 u}{2} - \frac{\tilde{A}}{2\tilde{c}}, \; -\frac{b_1 v}{2} - \frac{\tilde{B}}{2\tilde{c}}, \; \frac{b1}{2} \right), \tag{5.101}$$

$$\mathbf{L}^{(2)} = \left(\frac{1 - b_2}{2} - \frac{\hat{v}}{2\tilde{c}}, \; \frac{b_1 u}{2} - \frac{\tilde{B}}{2\tilde{c}}, \; \frac{b_1 v}{2} + \frac{\tilde{A}}{2\tilde{c}}, \; -\frac{b_1}{2} \right), \tag{5.102}$$

$$\mathbf{L}^{(3)} = \left(\frac{1 - b_2}{2} + \frac{\hat{v}}{2\tilde{c}}, \; \frac{b_1 u}{2} + \frac{\tilde{B}}{2\tilde{c}}, \; \frac{b_1 v}{2} - \frac{\tilde{A}}{2\tilde{c}}, \; -\frac{b_1}{2} \right), \tag{5.103}$$

$$\mathbf{L}^{(4)} = \left(\frac{b_2}{2} - \frac{\hat{u}}{2\tilde{c}}, \; -\frac{b_1 u}{2} + \frac{\tilde{A}}{2\tilde{c}}, \; -\frac{b_1 v}{2} + \frac{\tilde{B}}{2\tilde{c}}, \; \frac{b1}{2} \right), \tag{5.104}$$

$$\mathbf{R}^{(1)} = \begin{bmatrix} 1 \\ u - \tilde{A}\tilde{c} \\ v - \tilde{B}\tilde{c} \\ H - \hat{u}\tilde{c} \end{bmatrix}, \quad \mathbf{R}^{(2)} = \begin{bmatrix} 1 \\ u - \tilde{B}\tilde{c} \\ v + \tilde{A}\tilde{c} \\ H - \frac{1}{b_1} + \hat{v}\tilde{c} \end{bmatrix}, \tag{5.105}$$

$$\mathbf{R}^{(3)} = \begin{bmatrix} 1 \\ u + \tilde{B}\tilde{c} \\ v - \tilde{A}\tilde{c} \\ H - \frac{1}{b_1} - \hat{v}\tilde{c} \end{bmatrix}, \quad \mathbf{R}^{(4)} = \begin{bmatrix} 1 \\ u + \tilde{A}\tilde{c} \\ v + \tilde{B}\tilde{c} \\ H + \hat{u}\tilde{c} \end{bmatrix}, \tag{5.106}$$

where

$$q^2 = u^2 + v^2 \quad \hat{u} = \tilde{A}u + \tilde{B}v, \quad \hat{v} = \tilde{A}v - \tilde{B}u \tag{5.107}$$

$$b_1 = \frac{\gamma - 1}{\tilde{c}^2}, \quad b_2 = 1 + b_1 q^2 - b_1 H \tag{5.108}$$

$$\tilde{c} = \sqrt{\frac{\gamma p}{\rho}}. \tag{5.109}$$

5.9.3 2D Euler with Multiple Species

The 2D Euler equations with Multiple Species have $NS + 3$ equations thus implying that there will be NS+3 possibly non-distinct eigenvalues with associated eigenvectors. For the mixture of thermally perfect gases the enthalpy per unit mass as defined by equation (5.35) can be written as

$$h = \sum_{i=1}^{NS} Y_i h_i^f + \int_0^T c_p(s) ds, \tag{5.110}$$

and for a mixture of calorically perfect gases the above equation has the form

$$h = \sum_{i=1}^{NS} Y_i h_i^f + cpT. \tag{5.111}$$

Now considering the above equation we calculate the derivative of ρh with respect to the conserved variables

$$\frac{d(\rho h)}{d\rho} = h_{NS} + \rho c_p \frac{dT}{d\rho}, \tag{5.112}$$

$$\frac{d(\rho h)}{d(\rho u)} = \rho c_p \frac{dT}{d(\rho u)}, \tag{5.113}$$

$$\frac{d(\rho h)}{d(\rho v)} = \rho c_p \frac{dT}{d(\rho v)}, \tag{5.114}$$

$$\frac{d(\rho h)}{dE} = \rho c_p \frac{dT}{dE}, \tag{5.115}$$

$$\frac{d(\rho h)}{d(\rho Y_i)} = h_i - h_{NS} + \rho c_p \frac{dT}{(d\rho Y_i)}, \tag{5.116}$$

where i = 1 to $NS - 1$. These derivatives hold for both mixture of thermally perfect gases as well as for the mixture of calorically perfect gases. The equation (5.31) can be written as

$$p = -E + \frac{\rho(u^2 + v^2)}{2} + \rho h, \tag{5.117}$$

and the derivatives with respect to the conserved variables give

$$\frac{dp}{d\rho} = \frac{-(u^2 + v^2)}{2} + h_{NS} + \rho c_p \frac{dT}{d\rho}, \tag{5.118}$$

$$\frac{dp}{d(\rho u)} = u + \rho c_p \frac{dT}{d(\rho u)}, \tag{5.119}$$

$$\frac{dp}{d(\rho v)} = v + \rho c_p \frac{dT}{d(\rho v)}, \tag{5.120}$$

$$\frac{dp}{dE} = -1 + \rho c_p \frac{dT}{dE}, \tag{5.121}$$

$$\frac{dp}{d(\rho Y_i)} = h_i - h_{NS} + \rho c_p \frac{dT}{d(\rho Y_i)}, \tag{5.122}$$

where $i = 1$ to $NS - 1$. By taking the derivatives of equation (5.40) with respect to the conserved variables we obtain

$$\frac{dp}{d\rho} = R_{NS}T + \rho R \frac{dT}{d\rho}, \tag{5.123}$$

$$\frac{dp}{d(\rho u)} = \rho R \frac{dT}{d(\rho u)}, \tag{5.124}$$

$$\frac{dp}{d(\rho v)} = \rho R \frac{dT}{d(\rho v)}, \tag{5.125}$$

$$\frac{dp}{dE} = \rho R \frac{dT}{dE}, \tag{5.126}$$

$$\frac{dp}{d(\rho Y_i)} = (R_i - R_{NS})T + \rho R \frac{dT}{\rho Y_i}. \tag{5.127}$$

The above derivatives can be used to eliminate the derivatives of T in equations(5.118-5.122) and finally we will get

$$\frac{dp}{d\rho} = (\gamma - 1)(\frac{u^2 + v^2}{2} - h_{NS} + \frac{c_p R_{NS} T}{R}), \tag{5.128}$$

$$\frac{dp}{d(\rho u)} = (\gamma - 1)(-u), \tag{5.129}$$

$$\frac{dp}{d(\rho v)} = (\gamma - 1)(-v), \tag{5.130}$$

$$\frac{dp}{dE} = (\gamma - 1), \tag{5.131}$$

$$\frac{dp}{d(\rho Y_i)} = (\gamma - 1)(-h_i + h_{NS} + \frac{c_p (R_i - R_{NS})T}{R}), \tag{5.132}$$

which we will need while calculating the Jacobian matrix.

The Jacobian matrix is given by

$$\hat{u}I + JF + JB, \tag{5.133}$$

where

$$JF = \left(\frac{dp}{d\rho}\mathbf{J}_f \;\; \frac{dp}{d(\rho u)}\mathbf{J}_f \;\; \frac{dp}{d(\rho v)}\mathbf{J}_f \;\; \frac{dp}{dE}\mathbf{J}_f \;\; \frac{dp}{d(\rho Y_1)}\mathbf{J}_f \cdots \frac{dp}{d(\rho Y_{NS-1})}\mathbf{J}_f \right), \tag{5.134}$$

and

$$JB \;=\; (1\mathbf{J}_b \;\; u\mathbf{J}_b \;\; v\mathbf{J}_b \; H\mathbf{J}_b \;\; Y_1\mathbf{J}_b \cdots \;\; Y_{NS-1}\mathbf{J}_b)^T \,, \qquad (5.135)$$

where I is the $NS + 3$ by $NS + 3$ identity matrix, and

$$\mathbf{J}_f = \begin{bmatrix} 0 \\ \tilde{A} \\ \tilde{B} \\ \hat{u} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{J}_b = \begin{bmatrix} -\hat{u} \\ \tilde{A} \\ \tilde{B} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \qquad (5.136)$$

Let us write the eigenvalues of this Jacobian matrix as follows:

$$\lambda_1 \;=\; \hat{u} \;-\; \tilde{c}, \qquad (5.137)$$

$$\lambda_2 \;=\; \cdots \;=\; \lambda_{NS+2} \;=\; \hat{u}, \qquad (5.138)$$

$$\lambda_{NS+3} \;=\; \hat{u} \;+\; \tilde{c}. \qquad (5.139)$$

The rows of the following matrix are the left eigenvectors $\mathbf{L}^{(\tilde{p})}$

$$\begin{pmatrix} \frac{b_2}{2} + \frac{\hat{u}}{2\tilde{c}} + \frac{b_3}{2} & -\frac{b_1 u}{2} - \frac{\tilde{A}}{2\tilde{c}} & -\frac{b_1 v}{2} - \frac{\tilde{B}}{2\tilde{c}} & \frac{b1}{2} & \frac{-b_1 z_1}{2} & \cdots & \frac{-b_1 z_{NS-1}}{2} \\ 1 - b_2 - b_3 & b_1 u & b_1 v & -b_1 & b_1 z_1 & \cdots & b_1 z_{NS-1} \\ \hat{v} & \tilde{B} & -\tilde{A} & 0 & 0 & \cdots & 0 \\ -Y_1 & 0 & 0 & 0 & & & \\ \vdots & \vdots & \vdots & \vdots & & I & \\ -Y_{NS-1} & 0 & 0 & 0 & & & \\ \frac{b_2}{2} - \frac{\hat{u}}{2\tilde{c}} + \frac{b_3}{2} & -\frac{b_1 u}{2} + \frac{\tilde{A}}{2\tilde{c}} & -\frac{b_1 v}{2} + \frac{\tilde{B}}{2\tilde{c}} & \frac{b1}{2} & \frac{-b_1 z_1}{2} & \cdots & \frac{-b_1 z_{NS-1}}{2} \end{pmatrix}, \qquad (5.140)$$

while right eigenvectors $\mathbf{R}^{(\tilde{p})}$ are the columns of the following matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 1 \\ u - \tilde{A}\tilde{c} & u & \tilde{B} & 0 & \cdots & 0 & u + \tilde{A}\tilde{c} \\ v - \tilde{B}\tilde{c} & v & -\tilde{A} & 0 & \cdots & 0 & v + \tilde{B}\tilde{c} \\ H - \hat{u}\tilde{c} & H - \frac{1}{b_1} & -\hat{v} & z_1 & \cdots & z_{NS-1} & H + \hat{u}\tilde{c} \\ Y_1 & Y_1 & 0 & & & & Y_1 \\ \vdots & \vdots & \vdots & & I & & \vdots \\ Y_{NS-1} & Y_{NS-1} & 0 & & & & Y_{NS-1} \end{pmatrix}, \qquad (5.141)$$

where I is $NS-1$ by $NS-1$ identity matrix and

$$q^2 = u^2 + v^2, \quad \hat{u} = \tilde{A}u + \tilde{B}v, \quad \hat{v} = \tilde{A}v - \tilde{B}u, \tag{5.142}$$

$$\tilde{c} = \sqrt{p_\rho + \frac{pp_e}{\rho^2}}, \quad H = \frac{E + p}{\rho}, \tag{5.143}$$

$$b_1 = \frac{p_e}{\rho\tilde{c}^2}, \quad b_2 = 1 + b_1 q^2 - b_1 H, \tag{5.144}$$

$$b_3 = b_1 \sum_{i=1}^{NS-1} Y_i z_i, \quad z_i = -\frac{-p_{Y_i}}{p_e}, \tag{5.145}$$

where p_{Y_i} is the derivatives of pressure with respect to mass fraction of ith species.

5.10 Numerical Method

We have considered 1D case of equation (5.29) with no viscous effect,excluding and including the source terms by using the method of lines. The standard computational procedure followed in the method of lines is that the hyperbolic term $[\mathbf{f}(\mathbf{u})]$ is discretized with the suitable discretization scheme and resulting ODEs system is integrated with the suitable time integration package.

Due to its popularity the second order upwind scheme together with the van Leer limiter [7] was used for the spatial discretization. The motivation was to use a simpler alternative to the very complicated scheme (ENO) used for the spatial discretization in ([29]). Another novelty we have introduced is that instead of controlling CFL number or the local error per step we have controlled the local error per unit step, as in previous work.

An additional stringent requirements imposed on the numerical methods is that it must be able to handle step gradients, shock and contact discontinuities, that may develop spontaneously and then persist in the flow. Classical numerical schemes had a tendency to either produce large spurious oscillations near steep gradients. An introductory discussion of these difficulties and method can be found in [84]. Among the various upwind methods, one of the most popular is the Roe's scheme, which was originally proposed for a perfect gas [65]. The complicated procedure concerning the extension of the above scheme to non-equilibrium chemistry has been discussed in [76], its failure on some problems have been discussed in [25]. Due to simplicity and better shock capturing properties [25] we have used the Marquina Flux scheme for both reacting and non-reacting cases [25]. For the 2D problem the same considerations can be applied to the flux in the other spatial dimension. Then remains only to solve the quasilinear form of the mass fraction equation

$$\begin{pmatrix} \rho Y_1 \\ \vdots \\ \rho Y_{NS-1} \end{pmatrix}_t + \begin{pmatrix} \rho u Y_1 \\ \vdots \\ \rho u Y_{NS-1} \end{pmatrix}_x = 0. \tag{5.146}$$

These equation have been solved equation by equation (scalarly) and it is evident that the upwind direction for the x-direction is u respectively and similarly can be extended to other direction. In this concern we have followed the approach, as described in [76], and we have that

$$\hat{f}_{j+\frac{1}{2}}(U^l, U^r) = Y^l f^+_{\rho u}(U^l, U^r) + Y^r f^-_{\rho u}(U^l, U^r), \tag{5.147}$$

where $f^+_{\rho u}(U^l, U^r)$ and $f^-_{\rho u}(U^l, U^r)$ have been estimated by the procedure described in Chapter 2, U^l and U^r have been approximated with the second order upwind scheme.

5.10.1 Numerical Results

Problem 1 This is the popular shock tube test problem of Sod[70] for the one-dimensional, time dependent Euler equations for ideal gases with $\gamma = 1.4$ and has an exact solution. The problem models the flow of a gas in a long tube following the sudden breakdown of a diaphram separating two initial gas states at pressure and densities. Its solution will contain simultaneously a shock wave, a contact discontinuity and an expansion fan. Hence it is an attractive problem to judge the performance of the method. The initial conditions are given by

$$\begin{aligned} \rho(x,0) &= 1, & m(x,0) &= 0, & e(x,0) &= 2.5, & \text{for x} &< 0.5, \\ \rho(x,0) &= 0.125, & m(x,0) &= 0, & e(x,0) &= 0.25, & \text{for x} &> 0.5, \end{aligned} \tag{5.148}$$

where m is the momentum, ρ is the density and e is the specific total energy. The length of the domain is 1 and final time is 0.2 and the boundary conditions are transmissive [84].

Problem 2 The 1D Euler Equation for multi-species flow without chemical reaction has been solved first. The molar ratio 2/1/7 of $H_2/O_2/Ar$ has been assumed with the further assumption that they are thermally perfect gases. The initial data for 1D shock tube problem is

$$T = 400K \qquad p = 8000\frac{J}{m^3}, \tag{5.149}$$

on the left and

$$T = 1200K, \qquad p = 80000\frac{J}{m^3}, \tag{5.150}$$

on the right. The above equations show that we know the temperature and pressure and need to know the density and total energy for the mixture of thermally perfect gases.

First we describe the procedure to evaluate the density of the mixture, which can be evaluated by making use of the equation of state for multi-species as given by equation (5.40). The equation of state shows that there is a need to evaluate the gas constant for the mixture of thermally perfect gases, which is given by equation (5.39). It involves the mass fraction of each species, which can been calculated according to the procedure described in Section (5.6.4).

The total energy E of the mixture for the 1D Euler equations can be evaluated with help of equation (5.31), and we write as follow:

$$E = -p + \frac{\rho u^2}{2} + \rho h. \tag{5.151}$$

In the above equation, the procedure to evaluate enthalpy per unit mass for mixture of thermally perfect gases has been described in Section (5.8).

The length of the domain is 10cm and time is $40\mu s$ and this example has been taken from [28] and the transmissive boundary conditions (see [84]) have applied

problem 3 Now, we have considered a one-dimensional shock tube test problem with chemistry as given in [28]. Consider a shock hitting a solid wall boundary and reflecting off. Then after a delay a reaction wave kicks in at the boundary. The reaction wave picks up stream and merges with the shock causing a split into 3 waves. From wall to outflow (left to right) these waves are a rarefaction, a contact discontinuity and a shock (see for detail Fedkiw [28]).

Consider the 1D Euler equations for multi-species flow with chemistry (see appendix (C) for chemical mechanism). Similarly to Fedkiw [28] we have taken 2/1/7 molar ration of H2/O2/Ar and all the gases are assumed to be thermally perfect.

Now consider the initial data

$$\rho = .072\frac{kg}{m^3}, \qquad u = 0\frac{m}{s}, \tag{5.152}$$

$$p = 7173\frac{J}{m^3}, \tag{5.153}$$

on the left and on the right the initial data is given by

$$\rho \;=\; .1870\frac{kg}{m^3} \quad u \;=\; -487.34\frac{m}{s}, \tag{5.154}$$

$$p \;=\; 35594\frac{J}{m^3}, \tag{5.155}$$

and for details see Fedkiw [28]. The procedure to calculate the total energy has described in Problem 2 and for mass fraction calculation see Section (5.6.4). The length of domain is 10cm and time is 230μ s. The left side boundary conditions are reflective (see appendix (D) and [84]), while transmissive boundary conditions have been implemented on right hand side of the domain [84].

5.10.2 Results Discussion

The first Problem is a mild test problem and its exact solution has computed in the spatial domain $0 \leq\ x\ \leq 1$. The numerical solution is computed with $N = 140$ cells of constant meshsize. The second order upwind order together with the van Leer Limiter [7] has been used as the spatial discretization. The theta method [9] together with Gauss Seidel iterative method has been used as the time integration.

The novel technique we have introduced is that we have controlled the local error per unit step(see equation (5.2)) rather than the CFL number or the local error per step (see equation (5.1)). The results with LEPUS control strategy has been displayed in Figure (5.9), and code took 319 number of steps with the balancing factor (ϵ=0.3) with Gauss Seidel iterative method.

The results have also been obtained with LEPS control strategy for RTOL=0.1 and 0.01 and ATOL=10^{-5} and are displayed in Figures (5.10) and (5.11). The code took 383 and 566 number of steps with Gauss Seidel iterative method. The comparison to the exact solution shows that with LEPUS control strategy, same trend of accuracy follow as given by LEPS control strategy for RTOL=0.01 with less number of steps

The second Problem concerns with the modelling of the thermally perfect gases. So it is a hard problem and it is useful to assess the performance of the numerical technique. We have taken 200 cells and the convection term has been discretized with the second order upwind scheme and the theta method in combination with Gauss Seidel method as the time integrator [9].

The code takes 458 time steps with the local error per step control strategy to reach the final time with the balancing factor ($\epsilon = 0.3$) and the results are presented in Figure (5.12) which are comparable with the results as obtained by Fedkiw [29]. The numerical experiments have also been performed with higher value of the balancing factor, and when ϵ was greater that 0.5, then aacuracy started degrading when comparison was made to Fedkiw [28] results. In Figure (5.13) the results have been displayed with balancing factor $\epsilon = 0.6$ and the results are less accurate as compared to results obtained with $\epsilon = 0.3$ as given in figure (5.12).

The numerical experiments have also been performed with local error per step with (RTOL=0.1 and ATOL=10^{-4}). In this task the code took 482 steps and small oscillations are visible in the solutions the results are displayed in Figure (5.14), and it is evident that are small oscillations in the graph showing the density . The code has been run at the tighter RTOL =0.01 with same ATOL, and the results have been shown in Figure (5.15). The oscillations are no longer visible but the Code took 846 time steps as compared to the 458 time step with LEPUS control strategy.

For the comparison of both we have displayed the step size history with both strategies in Figure (5.16) and Figure (5.17) and which reveal that with local error per step control strategy the time-step is varying in oscillation manner while with local error per unit step control strategy the step-size is almost constant which is corresponding CFL number ($\frac{\overline{\triangle t}}{\overline{\triangle x}}$, where $\overline{\triangle t}$ and $\overline{\triangle x}$ are scaled time step and mesh size) 0.4 as given in Figure (5.18). Hence it can be concluded that with LEPUS control strategy equation (5.2) yields solutions at least as accurate as those obtained when controlling the LEPS control strategy chosen in order that the spatial discretization error is dominates.

The Problem 3 is a very hard problem in which the species continuity equations have been modelled along with 1D Euler equations, and has been taken from Fedkiw [28]. The domain has been discretized into 400 equally spaced grid cells. For spatial discretization we have used the 2nd order upwind method together with van Leer limiter [7] in contrary to approach adopted in Fedkiw[28], i.e., ENO schemes. The theta method together with Gauss Seidel iterative method has been used for the time integration

The code took a long time to run, and results show that there is small oscillation due to fact the numerical method has to resolve a one cell thick shock. The results with new technique LEPUS control strategy (see equation (5.2)) are given in Figure (5.19) for time of $230\mu s$ and the code took 6532 steps and the results are comparable to Fedkiw[28].

The code has also been run with local error per step control strategy as given by equation (5.1) with different RTOL and different ATOL. With RTOL= 0.1 and ATOL=1×10^{-4}, in this case, the code stop working after some time due to negative pressure being generated near the boundary and consequently a slightly tighter tolerance has been used and the code has been run with RTOL=0.1 and ATOL= 10^{-5}. In this case the code took 5549 steps to reach the final time=$230\mu s$ and got small oscillations as given in Figure (5.20). When the code had been run with RTOL=0.01 and ATOL=10^{-5}, the code took 8207 steps and the results are given in figure (5.21), again small oscillations are visible in this case also.

With both strategies, the results are of comparable accuracy except the species HO_2 and H_2O_2. When LEPS strategy is being used, the comparison of results to Fedkiw [28] have shown that for 0.1 relative tolerance HO_2 peak is higher and H_2O_2 peak is smaller and for 0.01 relative tolerance HO_2 peak is comparable to Fedkiw [28] but the peak of H_2O_2 is smaller. On the other hand when LEPUS strategy is being used HO_2 is exactly similar to Fedkiw [28] and H_2O_2 is slightly smaller than Fedkiw [28]. The shock position with both strategies is same.

From this we again draw the conclusion that LEPUS control strategy gives solution with the comparable accuracy to that of the LEPS control strategy, but for this extremely stiff and nonlinear problem, there is need to reduce the balancing factor ϵ and here we have chosen the balancing factor 0.025. The best choice of this parameter merits further investigation.

5.11 Conclusion

The method of lines has been used to solve the reactive flow (problems from atmospheric dispersion and combustion problems). This method reduces the partial differential equation to ordinary differential equations (ODEs) in time. The general procedure usually adopted in the time integration package is to control the local error per step. The new controlling strategy (LEPUS) based upon the error balancing approach has been successfully implemented instead of controlling the local error per step and the linearised equations have been solved with Guass Seidel iterative method.

A detailed description of the error balancing approach has been explained in Chapter 4 (see also [7, 49]). In the case of the atmospheric problem the new technique has been successfully implemented to both 1D and 2D. The efficiency of both strategies can be compared in Tables ((5.1) and (5.3)) comparing the number of the integration steps taken by the both strategies.

The framework and numerical results regarding combustion problems presented here shows that modern high accuracy numerical methods developed for gas dynamics can be usefully extended to the much more complicated problem of chemically reacting gas flows, and that these methods can effectively capture complex combustion phenomenon presented in these flows. The local error per unit step control strategy developed for the 1D Leveque and Yee problem [52] has been implemented to complex combustion phenomena and numerical results with Guass Seidel iterative method have revealed that with this control strategy the code is much more robust as compared to the controlling the local error per step. From this work it is clear that this technique merits further study.

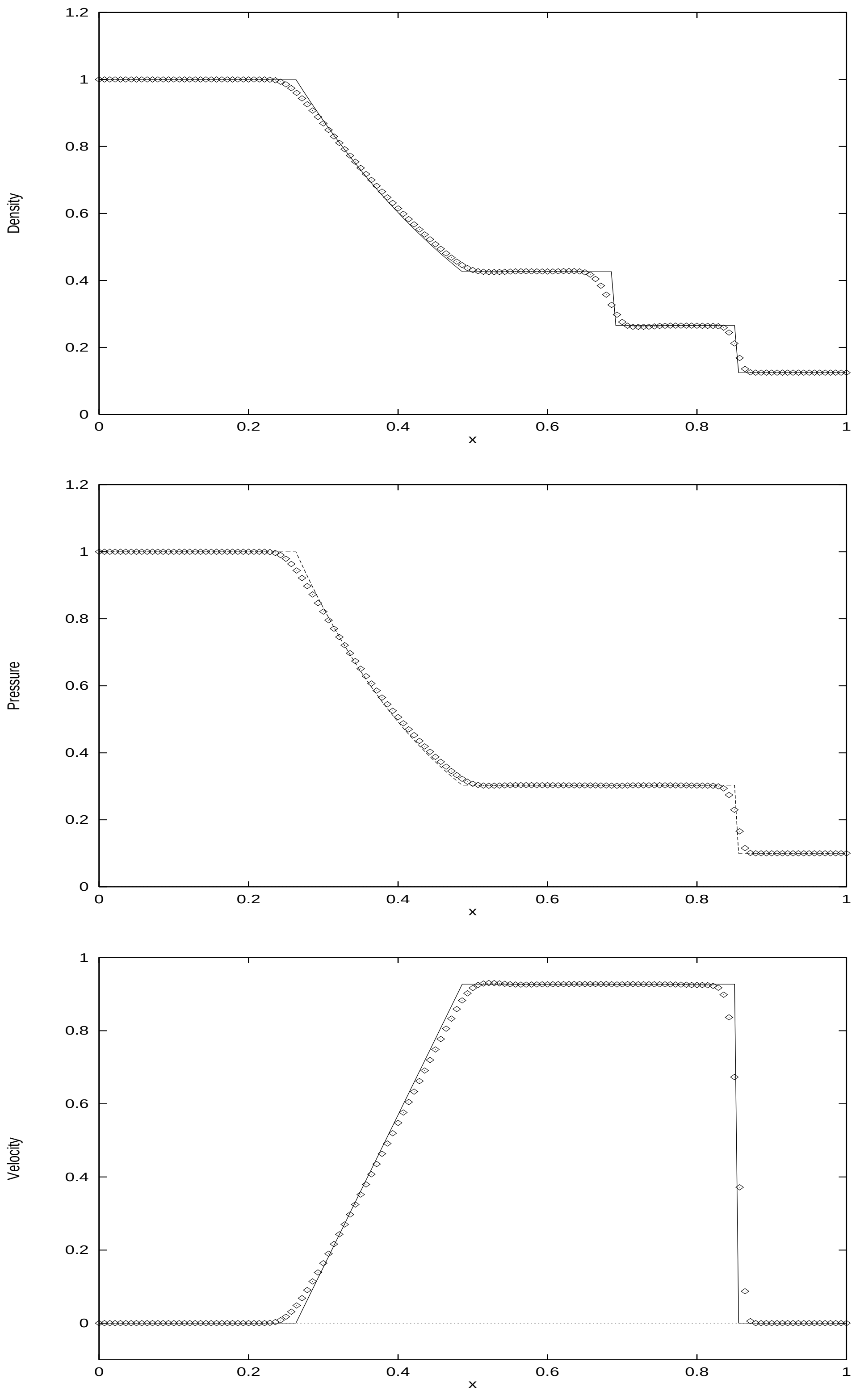


Figure 5.9: The numerical (dots) and exact (line) solutions of Problem 1 for the density, velocity and pressure using local error per unit step control strategy for balancing factor(ϵ) equal to 0.3 at time=0.2.

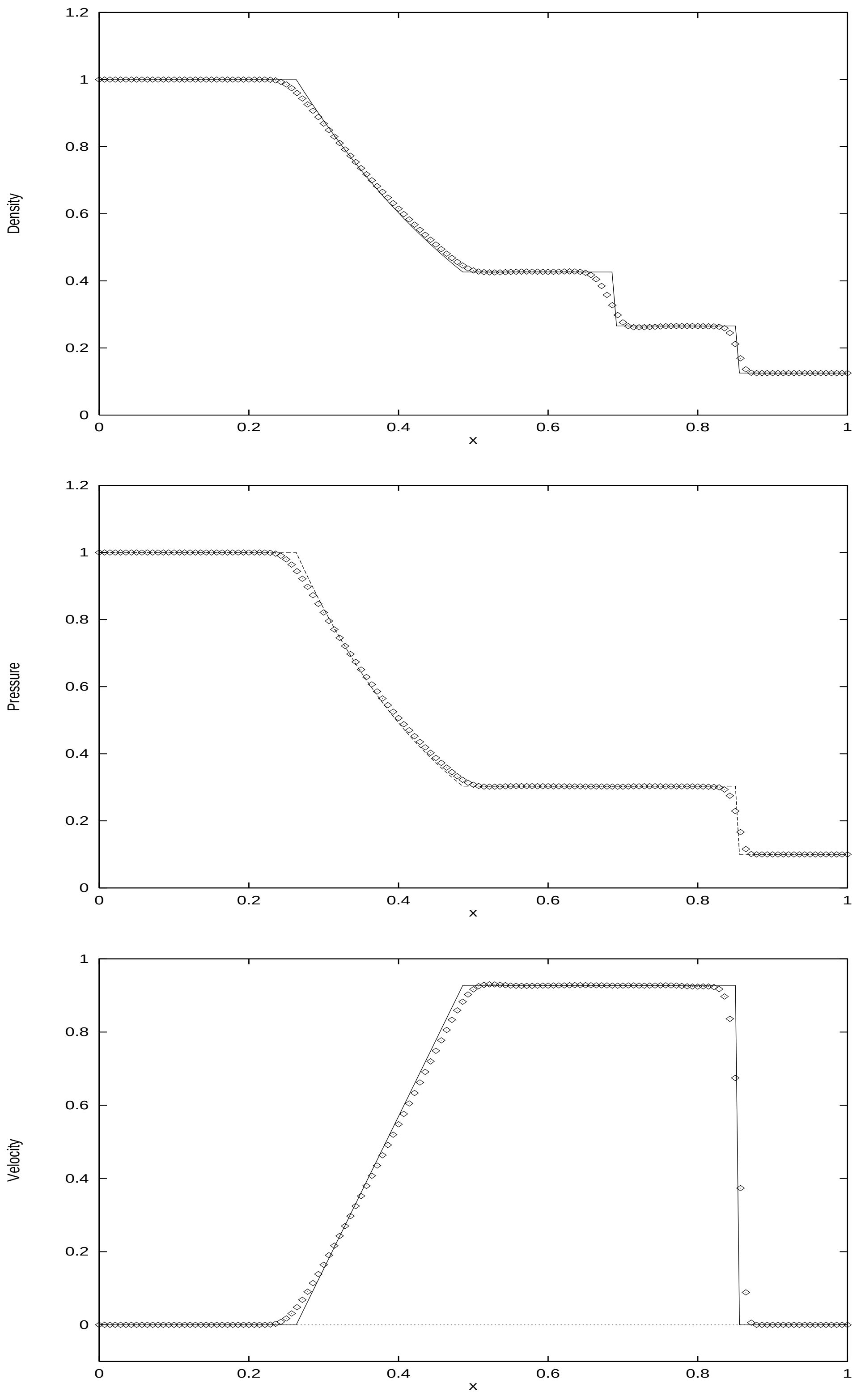


Figure 5.10: The numerical (dots) and exact (line) solutions of Problem 1 for the density, velocity and pressure using local error per step control strategy for 0.1 relative tolerance and 10^{-5} absolute tolerance at time=0.2.

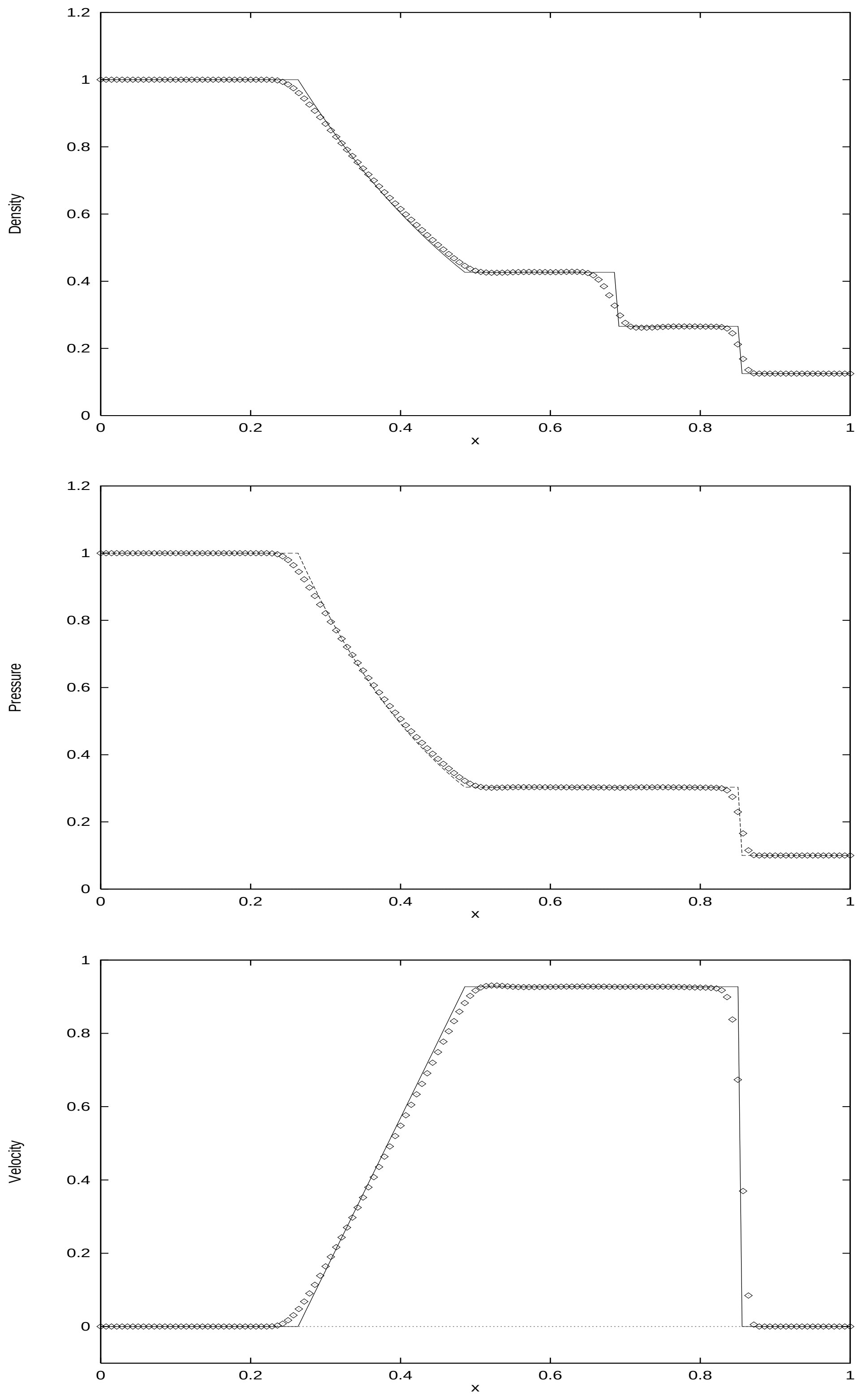


Figure 5.11: The numerical (dots) and exact (line) solutions of Problem 1 for the density, velocity and pressure using local error per step control strategy for 0.01 relative tolerance and 10^{-5} absolute tolerance at time=0.2.

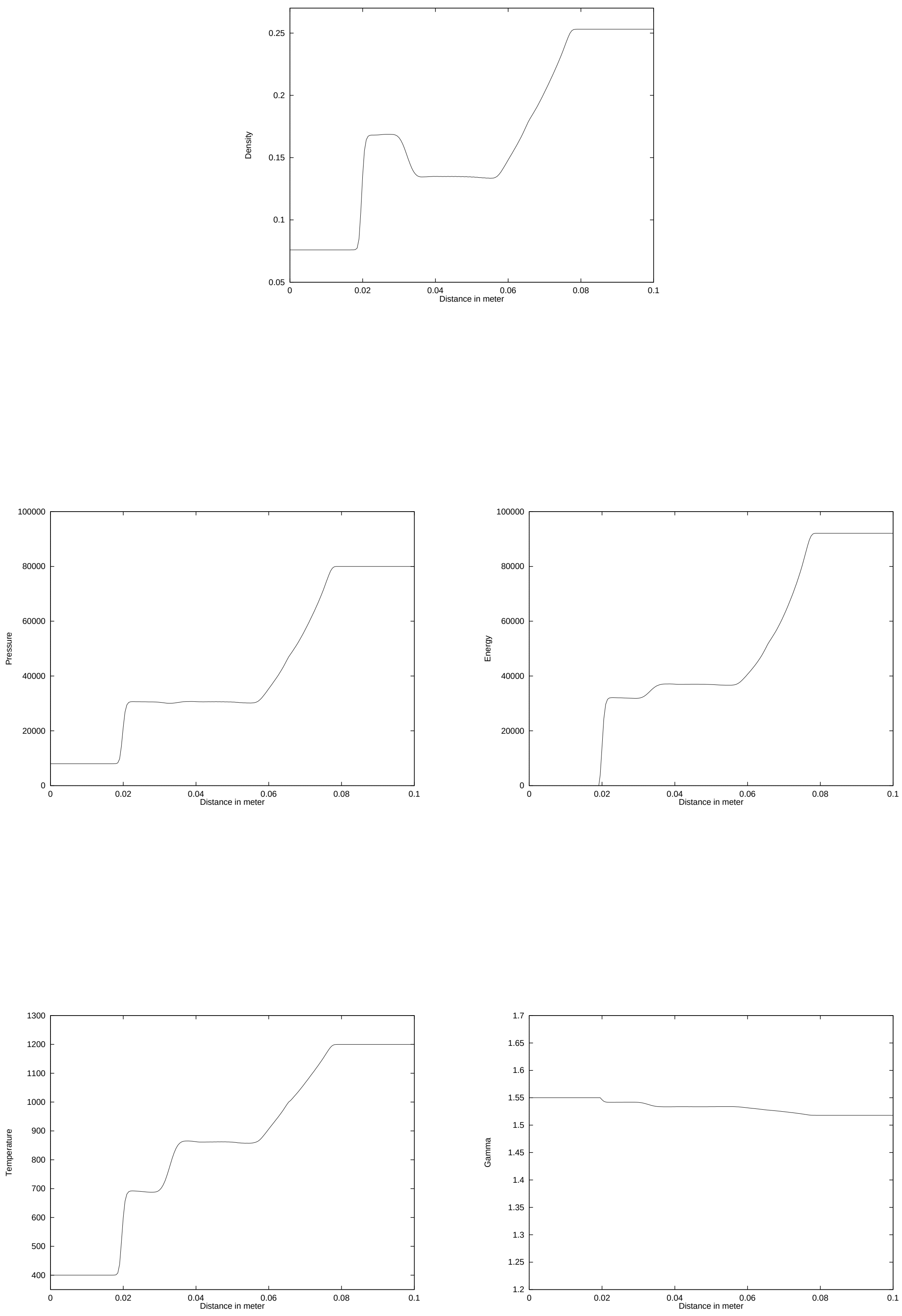


Figure 5.12: Numerical solution of Problem 2 using local error per unit step control strategy for balancing factor(ϵ) equal to 0.3 at time=40μs.

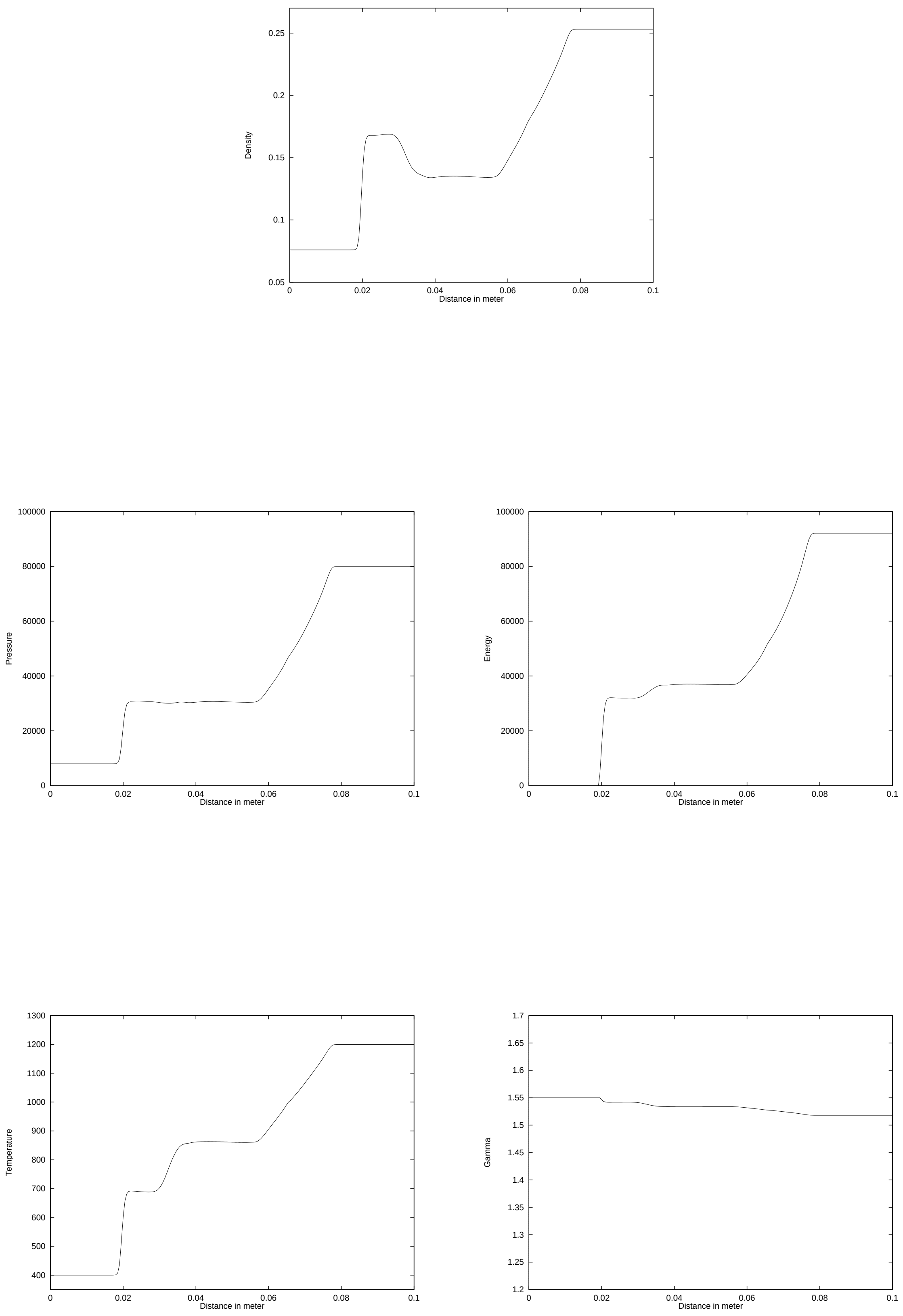


Figure 5.13: Numerical solution of Problem 2 with local error per unit step control strategy with for balancing factor (ϵ) equal to 0.6 at time=40μs.

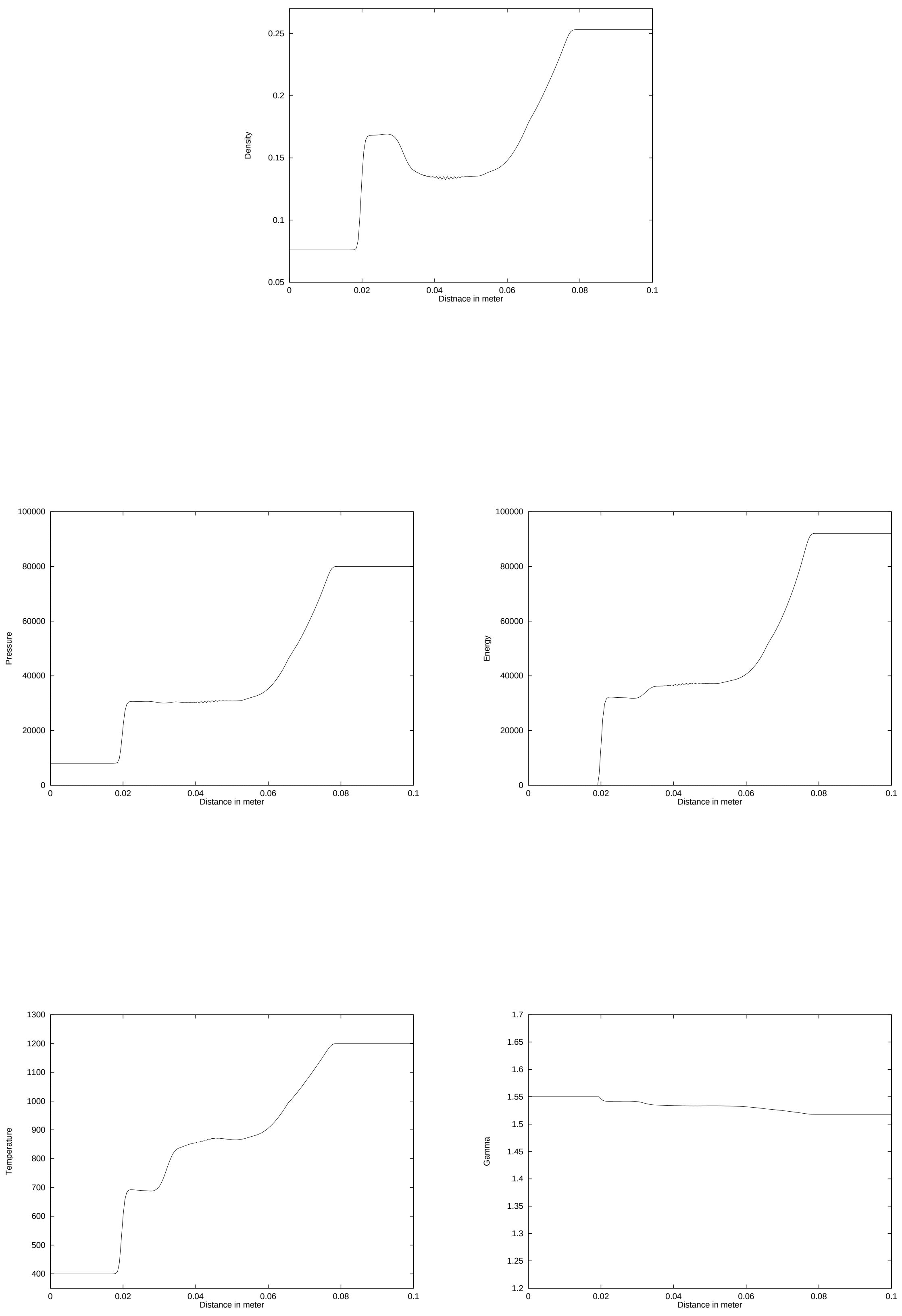


Figure 5.14: Numerical solution of Problem 2 using local error per step control strategy for 0.1 relative tolerance and 1×10^{-4} absolute tolerance at time=40μs.

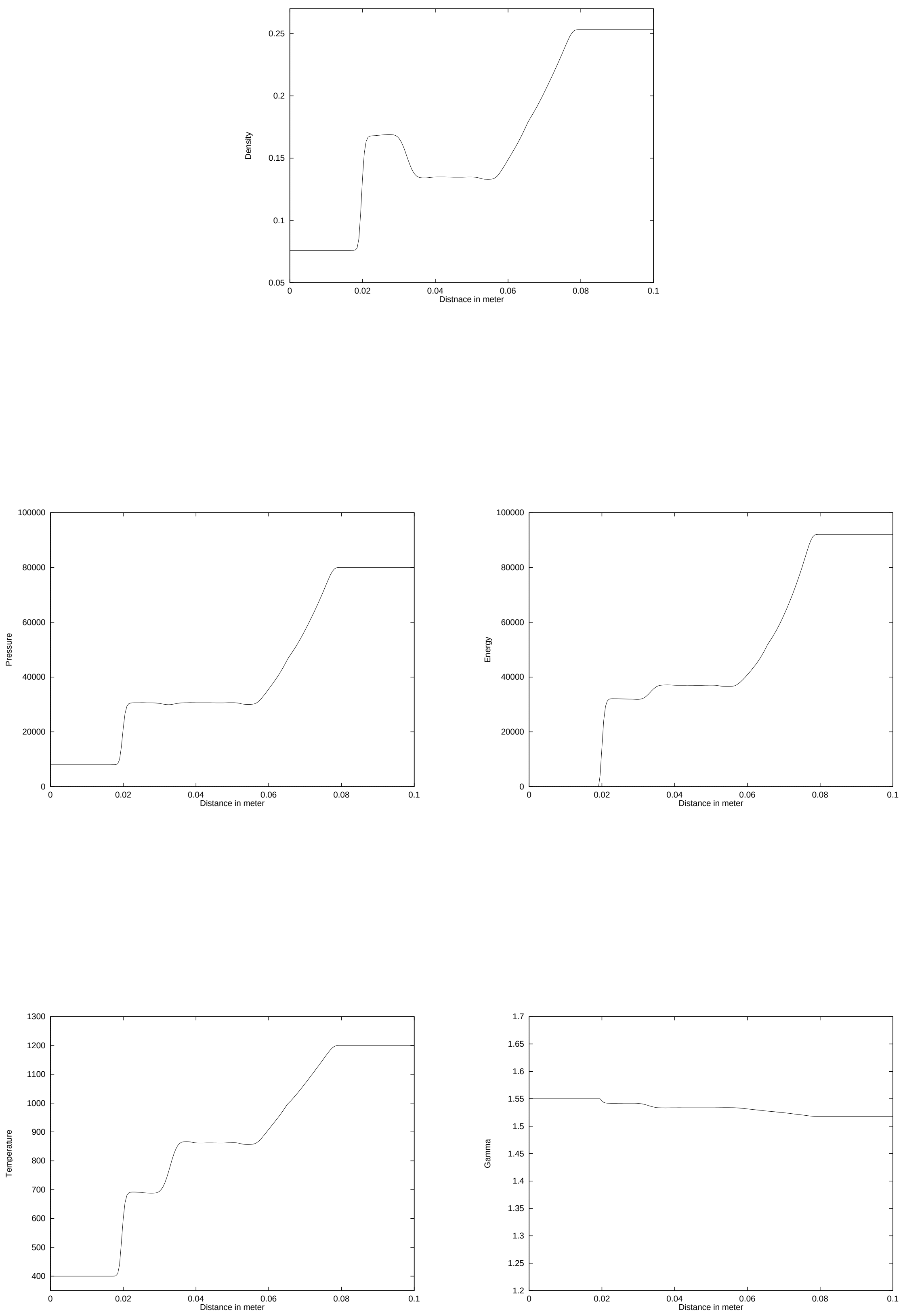


Figure 5.15: Numerical solution of Problem 2 using LEPS strategy for 0.01 relative tolerance and 1×10^{-4} absolute tolerance at time=40μs.

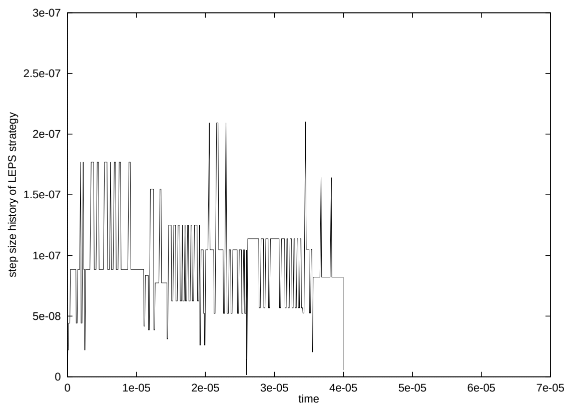


Figure 5.16: Step size history using LEPS strategy, for 0.1 relative tolerance and 1×10^{-4} absolute tolerance.

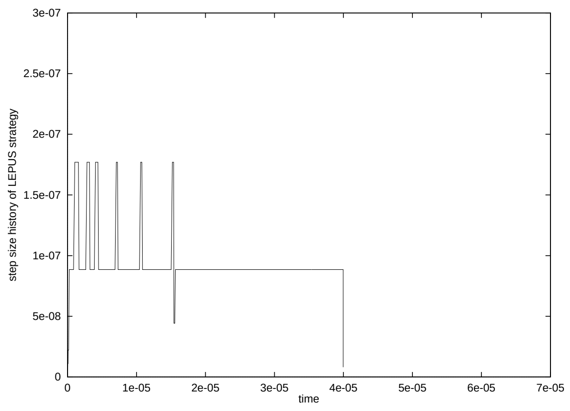


Figure 5.17: Step size history using LEPUS strategy for balancing factor(ϵ) equal to 0.3.

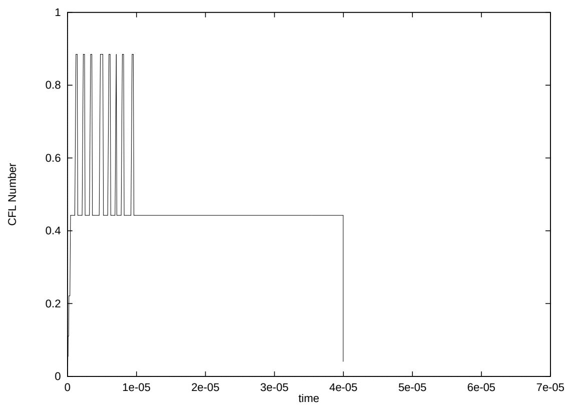


Figure 5.18: CFL number using LEPUS strategy for balancing factor(ϵ) equal to 0.3.

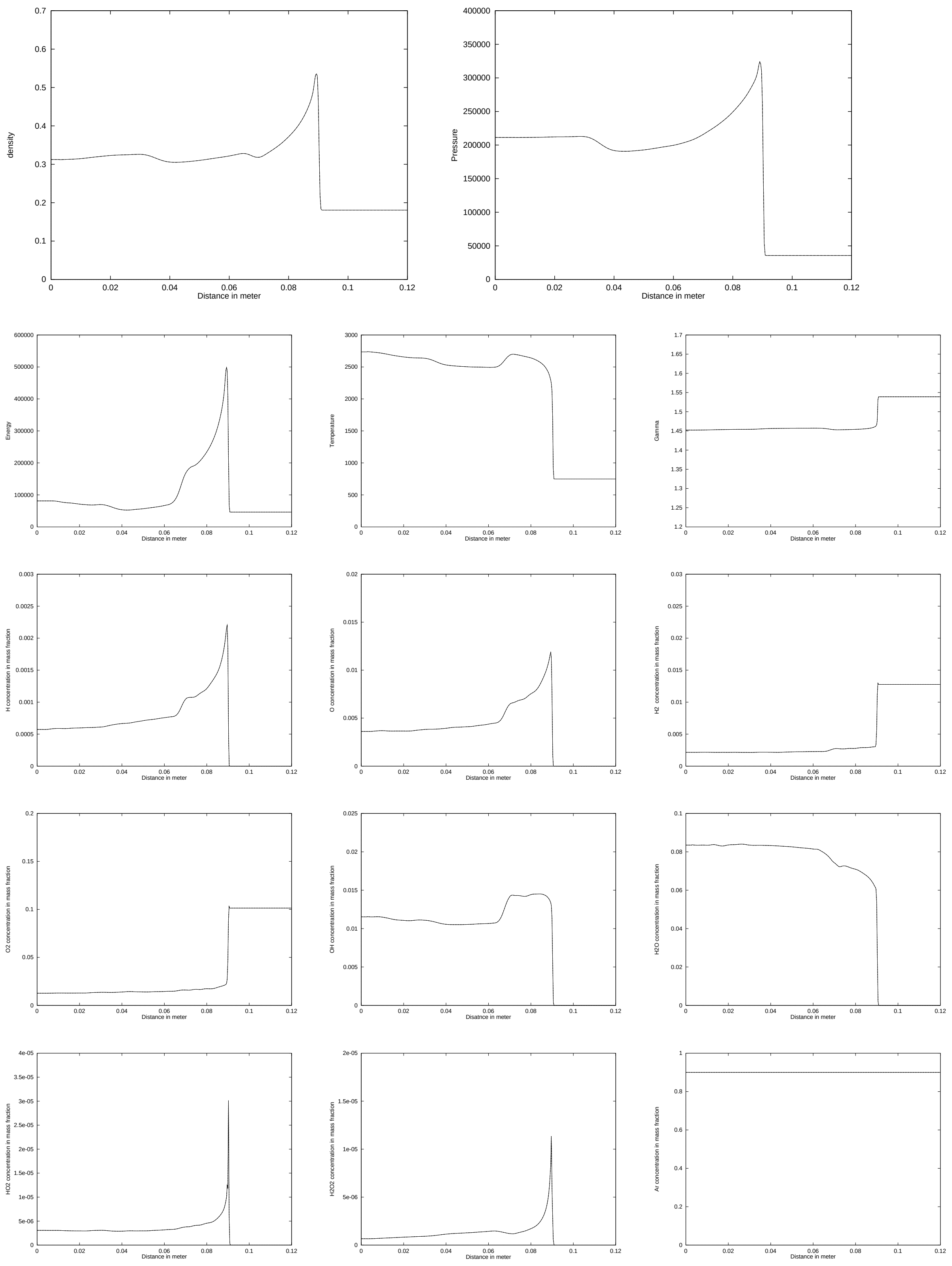


Figure 5.19: Numerical solution of Problem 3 with LEPUS control strategy for balancing factor(ϵ) equal to 0.025 at time=230μs.

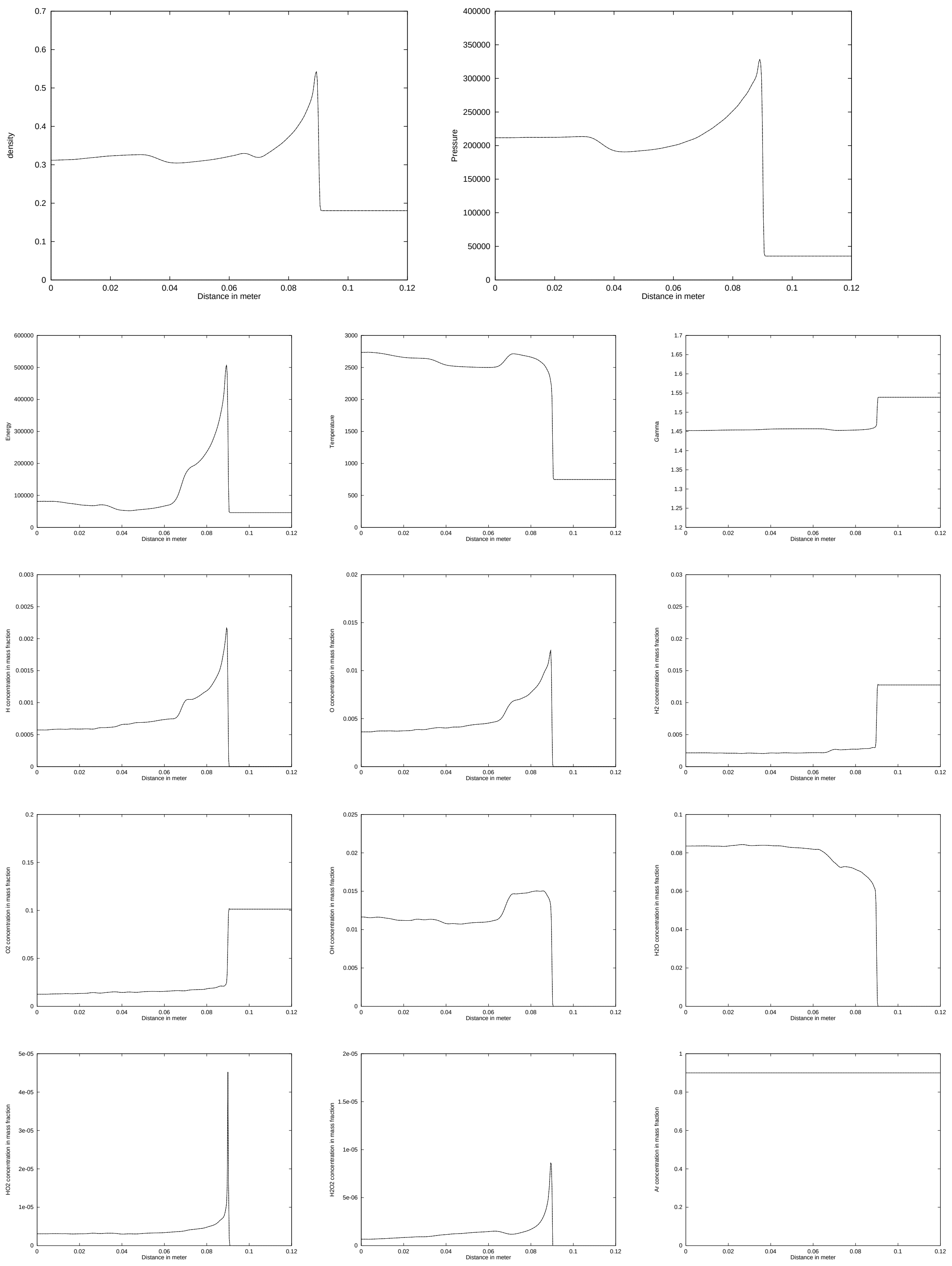


Figure 5.20: Numerical solution of Problem 3 with LEPS strategy for 0.1 relative tolerance and 1×10^{-5} absolute tolerance at time=230μs.

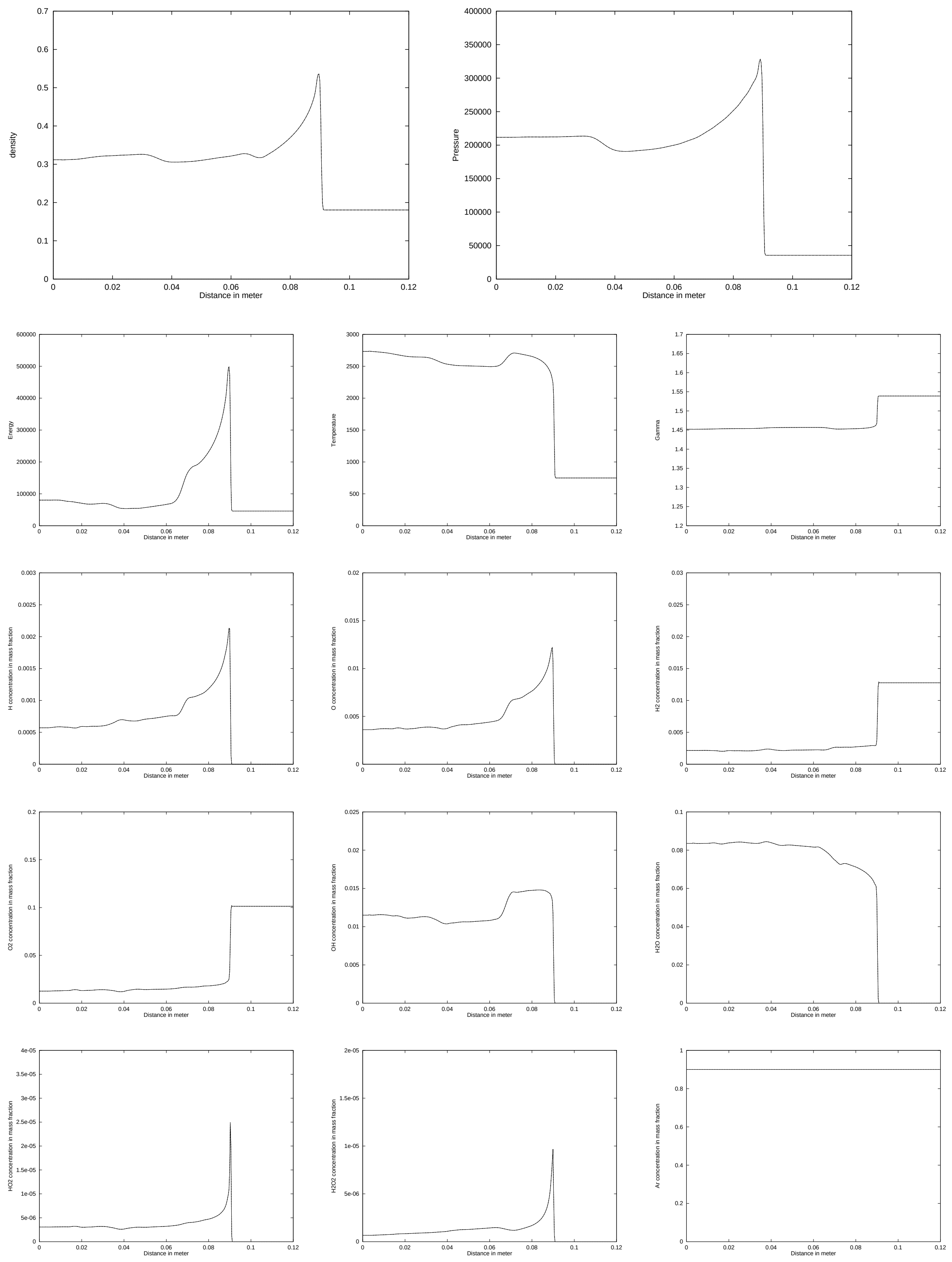


Figure 5.21: Numerical solution of Problem 3 with LEPS strategy for 0.01 relative tolerance and 1×10^{-5} absolute tolerance at time=230μs.

Chapter 6

Conclusion and Summary

The major aim of the present work is to develop an efficient, robust, highly accurate and general purpose numerical solver for reacting flow problems (such as atmospheric dispersion and combustion). In the research, attention has been paid to

- fast and efficient solvers for the chemical equations,
- space-time error balancing approach,
- local uniform grid refinement (1D case).

The first is of importance because the solution of reacting flow problems requires that stiff ODEs integration must be carried out at thousands of grid points many times in quick succession. This implies that the computational work is heavily dominated by the numerical treatment of the stiff odes system describing the chemical kinetics model in use. Hence it is of practical interest to investigate special purpose solvers which run faster without sacrificing accuracy and reliability.

For this reason ODE solver based upon the NDF2 method of Klopfenstein [47] has been developed for the solution of chemical kinetics arising from atmospheric chemistry. Analysis [3] has shown that the the stepsize integrator has increased by the factor of 1.26 over the same error in comparison with the BDF method. The NDF2 algorithm is second order which implies that there is no increase in storage requirements, and little increase in the computing effort.

It has formed a useful basis for constructing the numerical solution of the chemical kinetics arising from atmospheric chemistry in combination with the Gauss-Seidel iteration. The additional advantage of the Gauss-Seidel technique is that it reduces the storage requirements considerably. Also the storage requirement is the restrictive factor because in large air pollution models the chemistry has to be carried out at thousands of grid points.

Even though the NDF2 code in combination with Gauss Seidel showed improved performance as compared to Verwer's method [90], the results have indicated that some tuning of the stepsize strategy and Jacobian evaluation criteria may be needed. For this reason the theta method code has still proved useful, so we have used the theta method [9] as the time integrator to find the solution of model problem (1D and 2D Leveque and Yee problem [52]) and reacting flow problems.

The IMEX schemes [4] are widely used for the solution of the convection-diffusion equation. The effect of using the NDF2 scheme has been explored and stability analysis shows that in this case an IMEX scheme is stable on the purely imaginary axis unlike the BDF2 method.

In the second part of the research much effort has been put into devising a reliable, fast and efficient solver for PDEs arising from atmospheric and combustion problems. The method of lines approach has been used for the treatment of advection which reduces the PDEs to a system of ordinary differential equations with implementation of the suitable discretization scheme. For this purpose a second order upwind scheme together with the Van leer limiter [7] has been applied for the discretization of the spatial derivative.

For method of lines to be used efficiently, it is important that the time integration error should not dominate the error due to the spatial discretization of the PDEs, otherwise temporal errror will remove the benefits of using a good spatial discretization scheme.

This is really difficult if the LEPS is controlled by the integrator, since the relationship between the ODE global error and the chosen accuracy is not clear [49]. Also the spatial accuracy may vary with time, so any fixed tolerance used in the ODE integrator is unlikely to be related to the size of the changing spatial error. Hence we need a variable ODE tolerance which is related to the spatial discretization the error and can be modified when the spatial discretization error changes

So the new computational procedure based upon the error balancing approach of Berzins [7] has been extended to PDEs with source terms, the LEPUS is controlled rather than the LEPS and ODEs tolerance has varied with the local growth in time of the spatial discretization error. This strategy balanced the spatial and temporal errors, allowing the spatial discretization error to dominate. The optimum choice of the balancing factor is necessary so that the code may not take unnecessary time steps (blanacing factor too small) and accuracy may not be degraded (balancing factor too large) as explained in Chapter 4. In order to measure the local growth in time of spatial discretization error, we first calculated the primary solution using an upwind scheme and a quadrature rule to the source term, then the secondary solution is estimated an with an upwind scheme and quadrature rule, different to the previous one. The difference is the local growth in time of the spatial discretization error. The spatial error measured with the error balance approach has been used as the ODEs tolerance.

The starting point for the investigation of this approach has been the well-known problem of Leveque and Yee [52] and produced very promising results. We have observed that the local growth in time of the spatial discretization error is increased sharply near the discontinuity. In the case of this problem a lack of the spatial resolution yields a numerical front moving with the wrong speed because of the discontinuity present in the initial data. This needs a very fine mesh in the vicinity of the discontinuity, whereas a coarser mesh is adequate in the rest of the domain. Hence it is desirable to automatically adapt the mesh to follow the time-dependent nature of the solution which in turn requires some monitor function to adapt the grid. The commonly used monitor function, is based upon the spatial derivative. In this work we have introduced a new monitor function based upon the local growth in spatial discretization error and this has been successfully used to refine and coarsen the grid in a 1D case only.

In order to check the accuracy and reliability of the new approach, we have performed numerous tests and draw the conclusion that the LEPUS control strategy, gives as accurate results as that obtained with LEPS strategy.

After that we have tried to explore the feasibility of the new approach (LEPUS) on reactive flow problems. First we have solved the 1D and 2D atmospheric dispersion problems, again the results provide the evidence that LEPUS control strategy is as accurate as that of LEPS strategy. In the end we have tested a very hard problem related to combustion [29], in which the second order upwind in combination with van Leer limiter has been used instead of the complex ENO [29] scheme. We have attempted this problem both including and excluding chemistry, and in both case the LEPUS control strategy was applied. The results, even on this hard problem, have revealed that LEPUS control strategy has accuracy comparable to that of the LEPS control strategy.

Although LEPUS control strategy has worked well on almost all the problems, there is need to pay more attention regarding some unresolved issues stated below. The reason is that the local growth in time of spatial discretization error is increasing with increasing the stiffness of the source term. With large values of the local growth in spatial discretization error, the local error test made by the code is satisfied, but may lead to the convergence failure of the non-linear solver, and the code will take unnecessary steps.

Hence it is important that when the local growth in spatial discretization is large then the balancing factor ϵ should be small, which is the case only if we vary the balancing factor ϵ adaptively. The same is the case when the local growth in time of spatial discretization error is small, the code will take unnecessary steps. To avoid this problem we have taken a crude approximation, still there is more effort needed in this regard.

Although a second order upwind scheme together with the van Leer limiter were successfully implemented to a 1D combustion problem, still there are small oscillations in the solution, and the code is not much faster, hence there is need to attempt the use of a higher order upwind scheme.

Appendix A

Derivation of the Stability Equation

Here we will derive the equation (3.28), recall that characteristic polynomial is given by (see equation (3.27))

$$\lambda \triangle t = \tilde{q} = \xi + \frac{1}{2}\xi^2 - \hat{\alpha}\tilde{\gamma}_q \xi^3, \tag{A.1}$$

where $\xi = 1 - \lambda^{-1} = 1 - exp(-\iota\phi)$ and $-\pi \leq \phi \leq \pi$ and $\tilde{\gamma}_q = \sum_{m=1}^{q} \frac{1}{m}$ and for $q = 2$ $\tilde{\gamma}_q = \frac{3}{2}$, putting the ξ in equation (A.1) and simplification gives that

$$\lambda \triangle t = \frac{1}{2}(1 - exp(-\iota\phi))(3 - 3\hat{\alpha} - exp(-\iota\phi) + 6\hat{\alpha}exp(-\iota\phi) - 3\hat{\alpha}exp(-2\iota\phi)). \tag{A.2}$$

Using the fact that $exp(\iota\theta) = cos(\theta) + \iota sine(\theta)$ in the above equation with simplification gives that

$$\begin{aligned} \lambda \triangle t &= \frac{1}{2}(1 - cos(\phi))(3 - 3\hat{\alpha} - cos(\phi) + 6\hat{\alpha}cos(\phi) - 3\hat{\alpha}cos(2\phi)) \\ &+ \frac{1}{2}sin(\phi)(sin(\phi) - 6\hat{\alpha}sin(\phi) + 3\hat{\alpha}sin(2\phi)) \\ &+ \iota\frac{1}{2}sin(\phi)(3 - 3\hat{\alpha} - cos(\phi) + 6\hat{\alpha}cos(\phi) - 3\hat{\alpha}cos(\phi)) \\ &+ \frac{1}{2}\iota(1 - cos(\theta))((sin(\phi) - 6\hat{\alpha}sin(\phi) + 3\hat{\alpha}sin(\phi)). \end{aligned} \tag{A.3}$$

We will consider the real part of the above equation, which is

$$\begin{aligned} Re(q) &= \frac{1}{2}(1 - cos(\phi))(3 - 3\hat{\alpha} - cos(\phi) + 6\hat{\alpha}cos(\phi) - 3\hat{\alpha}cos(2\phi)) \\ &- \frac{1}{2}sin(\phi)(sin(\phi) - 6\hat{\alpha}sin(\phi) + 3\hat{\alpha}sin(2\phi),) \end{aligned} \tag{A.4}$$

and simplification gives that

$$\begin{aligned} Re(q) &= \frac{1}{2}[3 - 4cos(\phi) + cos(2\phi) \\ &+ (-3\hat{\alpha} + 9\hat{\alpha}cos(\phi) - 9\hat{\alpha}cos(2\phi)) \\ &+ 3\hat{\alpha}cos(3\phi)]. \end{aligned} \tag{A.5}$$

Using the identities

$$\begin{aligned} sin(\alpha + \beta) &= sin(\alpha)cos(\beta) + cos(\alpha)sin(\beta), \\ cos(\alpha + \beta) &= cos(\alpha)cos(\beta) - sin(\alpha)sin(\beta), \\ cos(2\alpha) &= cos^2(\alpha) - sin^2(\alpha), \\ sin(2\alpha) &= 2sin(\alpha)cos(\alpha). \end{aligned} \tag{A.6}$$

Using the identies defined in equation (A.6) successively we have that

$$3 - 4cos(\phi) + cos(2\phi) = 8sin^4(\theta), \tag{A.7}$$

$$\begin{aligned} 3\hat{\alpha}cos(3\phi) &= 3\hat{\alpha} - 54\hat{\alpha}cos(\theta) + 48\hat{\alpha}sin^2(\theta) \\ &+ 96\hat{\alpha}cos^2(\theta)sin^4(\theta), \end{aligned} \tag{A.8}$$

and

$$\begin{aligned} (-3\hat{\alpha} + 9\hat{\alpha}cos(\phi) - 9\hat{\alpha}cos(2\phi) &= -3\hat{\alpha} + 54\hat{\alpha}sin^2(\theta) \\ &- 72\hat{\alpha}sin^4(\theta), \end{aligned} \tag{A.9}$$

where $\theta = \frac{\phi}{2}$. Adding above three equations we have that

$$Re(q) = 4sin^4(\theta)[(1 - 3\hat{\alpha} + 12\hat{\alpha}cos^2(\theta)], \tag{A.10}$$

which is the required expression.

Appendix B

Atmospheric Reaction Schemes

B.0.1 First Reaction Scheme

This reaction scheme with constant reaction rate has been taken from [90]. The units for the rate constants are min^{-1} for first order reactions and $ppm^{-1}min^{-1}$ for the second order ones.

No.	reaction mechanism			rate constant
1.	NO_2	$\rightarrow$	$NO + O^3P$	$0.350 \times 10^{+00}$
2.	$NO + O_3$	$\rightarrow$	NO_2	$0.266 \times 10^{+02}$
3.	$HO_2 + NO$	$\rightarrow$	$NO_2 + OH$	$0.120 \times 10^{+05}$
4.	$HCHO$	$\rightarrow$	$2HO_2 + CO$	0.860×10^{-03}
5.	$HCHO$	$\rightarrow$	CO	0.820×10^{-03}
6.	$HCHO + OH$	$\rightarrow$	$HO_2 + CO$	$0.150 \times 10^{+05}$
7.	ALD	$\rightarrow$	$MEO2 + HO_2 + CO$	0.130×10^{-03}
8.	$ALD + OH$	$\rightarrow$	$C2O3$	$0.240 \times 10^{+05}$
9.	$C2O3 + NO$	$\rightarrow$	$MEO2 + NO_2 + CO_2$	$0.165 \times 10^{+05}$
10.	$C2O3 + NO_2$	$\rightarrow$	PAN	$0.900 \times 10^{+04}$
11.	PAN	$\rightarrow$.	$C2O3 + NO_2$	0.220×10^{-01}
12.	$MEO2 + NO$	$\rightarrow$	$CH3O + NO_2$	$0.120 \times 10^{+05}$
13.	$CH3O$	$\rightarrow$.	$HCHO + HO_2$	$0.188 \times 10^{+01}$
14.	$NO_2 + OH$	$\rightarrow$	HNO_3	$0.163 \times 10^{+05}$
15.	O^3P	$\rightarrow$	O_3	$0.480 \times 10^{+07}$
16.	O_3	$\rightarrow$	O^1D	0.350×10^{-03}
17.	O_3	$\rightarrow$	O^3P	0.175×10^{-01}
18.	O^1D	$\rightarrow$	$2OH$	$0.100 \times 10^{+09}$
19.	O^1D	$\rightarrow$	O^3P	$0.444 \times 10^{+12}$
20.	$SO_2 + OH$	$\rightarrow$	$SO_4 + HO_2$	$0.124 \times 10^{+04}$
21.	NO_3	$\rightarrow$	NO	$0.210 \times 10^{+01}$
22.	NO_3	$\rightarrow$	$NO_2 + O^3P$	$0.578 \times 10^{+01}$
23.	$NO_2 + O_3$	$\rightarrow$	NO_3	0.474×10^{-01}
24.	$NO_2 + NO_3$	$\rightarrow$	N_2O_5	$0.178 \times 10^{+04}$
25.	N_2O_5	$\rightarrow$	$NO_2 + NO_3$	$0.312 \times 10^{+01}$

B.0.2 Second Reaction Scheme

This reactions schemes has been borrowed from [10], which has 8 species and 7 reactions. The unit for the mth order rate constant is (molecule cm^{-3})$^{1-m}$s^{-1}. The photolysis rate constants have been pararmeterised as a function of the solar zenith, giving a first-order rate constant.

O_3				Reaction Rates
1.	$ROC + h\nu$	$\rightarrow$	$RP + ROC$	jp[1] = 1000exp(-4710/T)jp[3]
2.	$RP + NO$	$\rightarrow$	NO_2	kr[2] = 3.7098$\times$10^{-12}exp(242/T)jp[3]
3.	$NO_2 + h\nu$	$\rightarrow$	$NO + O_3$	jp[3] = 1.45$\times$10^{-2}exp(-0.4 sec(θ)
4.	$NO + O_3$	$\rightarrow$	NO_2	kr[4] = 1.7886$\times$10^{-12}exp(-1370/T)
5.	$RP + RP$	$\rightarrow$	RP	kr[5] = 6.7673$\times$10^{-12}
6.	$RP + NO_2$	$\rightarrow$	SGN	kr[6] = 1.00 $\times$ 10^{-13}
7.	$RP + NO_2$	$\rightarrow$	$SNGN$	kr[7] = 1.00 $\times$ 10^{-13}

Appendix C

Chemical Mechanism

Species	Molecular Weight	Low Temp	High Temp
H	1.00797	300	5000
O	15.9994	300	5000
H_2	2.01594	300	5000
O_2	31.9988	300	5000
OH	17.0074	300	5000
H_2O	18.0153	300	5000
HO_2	33.0068	300	5000
H_2O_2	34.0147	300	5000
Ar	39.9480	300	5000

Units

Molecular Weight	$g \times mole^{-1}$
Temp(T)	$Kelvin$
Pre Exp(A)	$cm \times mole \times sec \times Kelvin$
Temp Exp ($\tilde{\beta}$)	(non)
Act Eng (E_a)	$cal \times mole^{-1}$

REACTION MECHANISM						
	Reactions			PRE EXP	TEMP EXP	ACT ENG
1.	$O_2 + H$	$\rightarrow$	$OH + O$	2.00E+14	0.000	1.6802E+04
2.	$OH + O$	$\rightarrow$	$O_2 + H$	1.46E+14	0.000	4.9702E+02
3.	$H_2 + O$	$\rightarrow$	$OH + H$	5.06E+04	2.67	6.2860E+03
4.	$OH + H$	$\rightarrow$	$H_2O + O$	2.24E+04	2.67	4.3980E+03
5.	$H_2 + OH$	$\rightarrow$	$H_2O + O$	1.00E+08	1.60	3.2980E+03
6.	$H_2O + H$	$\rightarrow$	$H_2 + OH$	4.45E+08	1.60	1.8435E+04
7.	$OH + OH$	$\rightarrow$	$H_2O + O$	1.50E+09	1.14	1.0000E+02
8.	$H_2O + O$	$\rightarrow$	$OH + OH$	1.51E+10	1.14	1.7122E+04
9.	$H + H + M$	$\rightarrow$	$H_2 + M$	1.80E+18	-1.00	0.0000E+00
10.	$H_2 + M$	$\rightarrow$	$H + H + M$	6.99E+18	-1.00	1.0423E+05
11.	$H + OH + M$	$\rightarrow$	$H_2O + M$	2.20E+22	-2.00	0.0000E+00
12.	$H_2O + M$	$\rightarrow$	$H + OH + M$	3.80E+23	-2.00	1.1936E+05
13.	$O + O + M$	$\rightarrow$	$O_2 + M$	2.90E+17	-1.00	0.0000E+00
14.	$O_2 + M$	$\rightarrow$	$O + O + M$	6.81E+18	-1.00	1.1864E+05
15.	$H + O_2 + M$	$\rightarrow$	$HO_2 + M$	2.30E+18	-0.80	0.0000E+00
16.	$HO_2 + M$	$\rightarrow$	$H + O_2 + M$	3.26E+18	-0.80	4.6816E+04
17.	$HO_2 + H$	$\rightarrow$	$OH + OH$	1.50E+14	0.00	1.0040E+03
18.	$OH + OH$	$\rightarrow$	$HO_2 + H$	1.33E+13	0.00	4.0225E+04
19.	$HO_2 + H$	$\rightarrow$	$H_2 + O_2$	2.50E+13	0.00	6.9300E+02
20.	$H_2 + O_2$	$\rightarrow$	$HO_2 + H$	6.84E+13	0.00	5.8102E+04
21.	$HO_2 + H$	$\rightarrow$	$H_2O + O$	3.00E+13	0.00	1.7210E+03
22.	$H_2O + O$	$\rightarrow$	$HO_2 + H$	2.67E+13	0.00	5.7964E+04
23.	$HO_2 + O$	$\rightarrow$	$OH + O_2$	1.80E+13	0.00	-4.0600E+02
24.	$OH + O_2$	$\rightarrow$	$HO_2 + O$	2.18E+13	0.00	5.5117E+04
25.	$HO_2 + OH$	$\rightarrow$	$H_2O + O_2$	6.00E+13	0.00	0.0000E+00
26.	$H_2O + O_2$	$\rightarrow$	$HO_2 + OH$	7.31E+14	0.00	7.2545E+04
27.	$HO_2 + HO_2$	$\rightarrow$	$H_2O_2 + O_2$	2.50E+11	0.00	-1.2430E+03
28.	$OH + OH + M$	$\rightarrow$	$H_2O_2 + M$	3.25E+22	-2.00	0.0000E+00
29.	$H_2O_2 + M$	$\rightarrow$	$OH + OH + M$	2.10E+24	-2.00	4.9426E+04
30.	$H_2O_2 + H$	$\rightarrow$	$H_2 + HO_2$	1.70E+12	0.00	3.7520E+03
31.	$H_2 + HO_2$	$\rightarrow$	$H_2O_2 + H$	1.15E+12	0.00	1.9331E+04
32.	$H_2O_2 + H$	$\rightarrow$	$H_2O + OH$	1.00E+13	0.00	3.5850E+03
33.	$H_2O + OH$	$\rightarrow$	$H_2O_2 + H$	2.67E+12	0.00	7.3497E+04
34.	$H_2O_2 + O$	$\rightarrow$	$OH + HO_2$	2.80E+13	0.00	6.4050E+03
35.	$OH + HO_2$	$\rightarrow$	$H_2O_2 + O$	8.40E+12	0.00	2.0098E+04
36.	$H_2O_2 + OH$	$\rightarrow$	$H_2O + HO_2$	5.40E+12	0.00	1.0040E+03
37.	$H_2O + HO_2$	$\rightarrow$	$H_2O_2 + OH$	1.63E+13	0.00	3.1718E+04

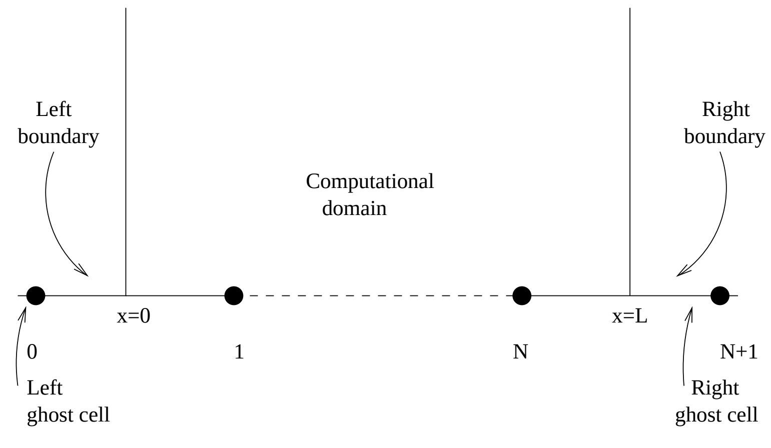
Collision efficiences in reactions with M:

$f_{H_2} = 100, \; F_{O_2} = 0.35, \; F_{H_2O} = 6.5 \;\; f_{N_2} = 0.5$

Appendix D

Boundary Conditions

The imposition of boundary condition is, fundamentally, a physical problem so great care is required in their implementation. For this consider the domain $[0,\ L]$, which has been discretized into N computing cells of length $\triangle x$ we need boundary conditions at the boundaries $x = 0$ and $x = L$ as shown in Figure(D). Numerically, boundary conditions provide the numerical fluxes $\mathbf{f}_{\frac{1}{2}}$ and $\mathbf{f}_{N+\frac{1}{2}}$. The boundary conditions may result in direct evaluation of fluxes $\mathbf{f}_{\frac{1}{2}}$ and $\mathbf{f}_{N+\frac{1}{2}}$. We may also have an alternative route, in which we have fictitious values in the ghost cells I_0 and I_{N+1}, adjacent to I_1 and I_N respectively. So boundary Riemann problems $RP(\mathbf{U}_0, \mathbf{U}_1)$ and $RP(\mathbf{U}_N, \mathbf{U}_{N+1})$ are solved and the corresponding fluxes $\mathbf{f}_{\frac{1}{2}}$ and $\mathbf{f}_{N+\frac{1}{2}}$ are computed with appropriate Riemann solver. Suppose that at $x = L$



reflective boundary conditions have been applied. Then fictitious state $\mathbf{U}_{N+1}(t_n)$ is defined from the known state $\mathbf{U}_N(t_n)$ inside the computational domain, namely

$$\rho_{N+1}(t_n) = \rho_N(t_n) \quad u_{N+1}(t_n) = u_N(t_n), \quad p_{N+1}(t_n) = p_N(t_n). \qquad \text{(D.1)}$$

In the above equation (D.1), ρ is the density u is the velocity, p is the pressure.

Bibliography

[1] A. Abgrall, L. Fezoui, and T. Talandier. An extension of Osher's Solver for chemical and vibrational non-equilibrium flows. *International Journal for Numerical Methods in Fluids*, 14:935–960, 1992.

[2] R. Abgrall. Preliminary results on the extension of Roe's Riemann solver to non-equilibrium flows. Technical Report 987, INRIA, 1989.

[3] I. Ahmad and M. Berzins. An algorithm for ODEs from atmospheric dispersion problems. *Applied Numerical Mathematics*, 25:137–149, 1997.

[4] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent PDEs. *SIAM Journal on Numerical Analysis*, 32(3):797–823., 1995.

[5] C. Basdevant, M. Deville, P. Haldenwang, J. Lacroix, J. Ouazzani R. Peyret P.Orlandi, and A. Patera. Spectral and finite difference solutions of the Burgers equation. *Comp. Fluids.*, 14:23–41, 1986.

[6] M. Berzins. A C^1 interpolant for codes based on backward differentiation formulae. *Applied Numerical Mathematics*, 2:109–118, 1986.

[7] M. Berzins. Temporal error control for convection-dominated equations in two space dimensions. *SIAM Journal on Scientific and Statistical Computing*, 16:558–580, 1995.

[8] M. Berzins, P. M. Dew, and R. M. Fuzerland. Developing software for time-dependent problems using the method of lines and differental algebraic integrators. *Applied Numerical Mathematics*, 5:375–397, 1989.

[9] M. Berzins and R.M. Fuzerland. An adaptive theta method for the solution of stiff and non-stiff differential equation. *Applied Numerical Mathematics*, 9:1 –19, 1992.

[10] M. Berzins, P. H. Gaskell, A. Sleigh, W. Speares, A. Tomlin, and J. M. Ware. An adaptive CFD solver for time dependent environmental flow problems. In *K.W. Morton and M. J. Baines, eds., Numerical Methods for Fluid Dynamics V*, pages 311–318. (Clarendon Press, Oxford), 1995.

[11] M. Berzins and J. M. Ware. Positive cell-centered finite volume discretization methods for hyperbolic equations on irregular meshes. *Applied Numerical Mathematics*, 16:417–438, 1995.

[12] M. Berzins and J. M. Ware. Solving convection and convection reaction problems using the M.O.L. *Applied Numerical Mathematics*, 20:83–99, 1996.

[13] L. Bledjian. Computation of time dependent laminar flame structure. *Combust. Flame*, 20:5–17, 1973.

[14] J. G. Blom, R. A. Trompert, and J. G. Verwer. *A Vectorizable Adaptive Grid Solver for PDEs in 2D.* CWI, P. O. Box 94079, 1090 GB Amsderdam, The Netherlands, 1996. NM-R3919.

[15] K. Brenan, S. Campbell, and L. R. Petzold. *Numerical solution of Initial Value Problems in Differential Algenraic Equatioations.* (SIAM, Philadelphia), 1996.

[16] R. L. Burden and J. D. Faires. *Numerical Analysis.* PWS Publishing Company Boston, 5th edition, 1993.

[17] T. S. Chua and P. M. Dew. The design of a variable-step integrator for the simulation of gas transmission networks. *International Journal for Numerical Methods in Engineering*, 20:1797–1813, 1984.

[18] P. Collela and H. M. Glaz. Efficient solution algorithm for the Riemann problem for real gases. *Journal of Computational Physics*, 59:264–289, 1985.

[19] P. Collela, A. J. Majda, and V. Roytburd. Theoretical and numerical structure of reacting shock waves. *SIAM Journal on Scientific and Statistical Computing*, 7:1059–1080, 1986.

[20] P. Collela and P. R. Woodward. The PPM method for gas dynamics simulation. *Journal of Computational Physics*, 54:174–201, 1984.

[21] R. B. I. Johnson B. J. Cory and M. J. Short. A tunable integration method for the simulation of power system dynamimics. IEEE/PE 88 Winter Meeting Paper 88 WM 177-8, 1988.

[22] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume II. (Interscience, New York), 1963.

[23] R. Courant, E. Issacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure. Appl. Math.*, 5:243–255, 1952.

[24] G. G. Dahlquist. A special stabiliity problem for linear multistep methods. *BIT*, 3:27–23, 1963.

[25] R. Donat and A. Marquina. Capturing shock reflections: An improved flux formula. *Journal of Computational Physics*, 125:42–58., 1996.

[26] B. Engquist and S. Osher. One sided difference approximations for nonlinear conservation law. *Mathematics of Computation*, 36:321–352, 1981.

[27] W. H. Enright, T.E. Hull, and B. Lindberg. Comparing numerical methods for stiff systems of ODEs. *BIT*, 15:10–48, 1975.

[28] R. P. Fedkiw. *A Survey of Chemically Reacting, Compressible Flow.* UCLA (Ph.D. Dissertation), 1966.

[29] R. P. Fedkiw, B. Merriman, and S. Osher. High accuracy numerical methods for thermally perfect gas flows with chemistry. *Journal of Computational Physics*, 132:175–190, 1997.

[30] R. P. Fedkiw and B. Merriman S. Osher. *Efficient Characteristic Projection in Upwind Difference Schemes for Hyperbolic systems.* CAM Report 97-17, 1997.

[31] R. M. Fuzerland. *The construction of adaptive space mesh.* TNER.85.022, Shell Research Ltd, Thornton Research Center, Chester England, 1985.

[32] S. Galant. *An Improved Method of Lines to Compute Time Dependent Laminar Flame Structure, Eighteenth Symposium (International) Combustion.* pp. 1451-1459, The Combustion Institute, Pittsburgh, PA, 1981.

[33] P. Glaister. An approximate linearised Riemann solver for the Euler equations for real gases. *Journal of Computational Physics*, 74:382–408, 1988.

[34] P. Glaister. Flux difference splitting for the Euler equations in one spatial co-ordinate with are variation. *International Journal for Numerical Methods in Fluids*, 8:97–119, 1988.

[35] S. K. Godnuov. A difference schemes for numerical solution of discontinuous solution of hydrodynamics equations. *Math. Sbornik*, 47:271–306, 1959. (in Russian).

[36] A. Harten. *High resolution scheme for hyperbolic conservation laws.* New York University Report DOE/ER 03077-175, 1982.

[37] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate ENO. schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[38] A. Harten and J. M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50:235–269, 1983.

[39] A. C. Heard, M. J. Pilling, and A. S. Tomlin. Mechanism reduction techniques applied to tropospheric chemistry. *Atmospheric Environment*, 32:1059–1073, 1998.

[40] A. C. Hindmarsh. ODEPACK - a systematized collection of ode solvers. In *Advances in Computer Methods IV*, New Brunswick , CT, 1983. IMACS, R. Vicchenevetsky and R. S. Stepleman, eds.

[41] T. R. Hopkins. *Numerical solution of quasi-linear parabolic PDEs.* PhD thesis, Liverpool University, 1976.

[42] J. Frank W. Hunsdorfer and J. G. Verwer. On the stability of implicit-explicit linear multistep methods. *Applied Numerical Mathematics*, 25:193–205, 1997.

[43] W. Hunsdorfer and J. G. Verwer. A note on splitting errors for advection reaction equations. *Applied Numerical Mathematics*, 18:191–199, 1995.

[44] A. Jameson. *A NonOscillatory Shock Capturing Scheme using Flux Limited Dissipation*, volume 22. Lectures in Applied Mathematics, 1985.

[45] R. J. Kee, F. M. Rupley, and J. A. Miller. The chemkin thermodynamic data base. Technical Report SAND87-8215B, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94551, April 1994.

[46] C. T. Kelly. *Iterative Methods for Linear and Nonlinear Equations.* (SIAM, Philadelphia), 1995.

[47] R.W. Klopfenstein. Numerical differentiation formulae for stiff systems of ODEs. *RCA Rev*, 32:447–462, 1971.

[48] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems, The Initial Value Problem.* John Wiley and Sons Ltd., England, April 1995.

[49] J. Lawson, M. Berzins, and P. M. Dew. Balancing space and time errors in the method of lines for parabolic equation. *SIAM Journal on Scientific and Statistical Computing*, 12(12):573–594, 1991.

[50] P. D. Lax. Hyperbolic systems of conservation laws and the mathematical theory of shock waves. Technical Report SIAM Philadelphia, Pennsylvania 19103., Courant Institute of Mathematical Sciences, New York University., 1973.

[51] P.D. Lax. Hyperbolic systems of conservation laws II. *Comm. Pure. Appl. Math.*, 10:537, 1957.

[52] R. J. Leveque and H.C. Yee. A study of numerical methods for hyperbolic conservation laws with stiff source terms. *Journal of Computational Physics*, 86:187–210, 1990.

[53] M.-S. Liou, B. van Leer, and J. S. Shuen. Splitting of inviscid flues for real gases. *Journal of Computational Physics*, 87:1–24, 1990.

[54] O. A. Liskovets. The methods of lines. *Differential 'nye Uravneniya 1: 1662-1678: English translation in Differ. Equations*, 1:1308–132, 1965.

[55] M. J. Maron. *Numerical Analysis, A Practical Approach.* Macmillan Publishing Co Inc., New York, 1982.

[56] S. H. Maron and C. F. Prutton. *Principles of Physical Chemistry.* The Macmillan Company, New York, Collier-Macmillan Limited, London, 4th edition, 1965.

[57] G. J. McRae, W. R. Goodin, and J. H. Seinfield. Numerical solution of the atmospheric diffusion equation for chemically flows. *Journal of Computational Physics*, 45:1–42, 1982.

[58] E. O. Oran and J. P. Boris. *Numerical Simulation of Reactive Flow.* Elsevier Science Publishing Co., 1987.

[59] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM Journal on Numerical Analysis*, 21(2):217–235, 1984.

[60] S. Osher and F. Solomon. Upwind differences schemes for hyperbolic systems of conservation laws. *Mathematics of Computation*, 38:339–374, 1982.

[61] M. V. Papalexandris, A. Leonard, and P. E. Dimotakis. Unsplit schemes for hyperbolic conservation laws with source terms in one space dimension. *Journal of Computational Physics*, 134:31–61, 1997.

[62] S. V. Pennington and M. Berzins. New NAG library software for first-order partial differential equations. *ACM Transactions on Mathematical Software*, 20:63–99, 1994.

[63] L. Petzold. Differential/ algebraic equations are not odes. *SIAM Journal on Scientific and Statistical Computing*, 3(3):367–384, 1982.

[64] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Mathematics of Computation*, 28:145., 1974.

[65] P. L. Roe. Approximate riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[66] P. L. Roe. Upwind differencing schemes for hyperbolic conservation laws with source terms. In *Lecture notes in Mathematics 1270, Nonlinear Hyperbolic*

problems. C. Carasso, P. A. Raviart. D. Serre. Eds, Springer-Verlag PP. 41-51, 1986.

[67] P. L. Roe. Finite-volume methods for the compressible Navier-Stoke equations. In *Proc. 5th International Conference on Laminar and Turbulent Flow,Montreal. 1987*, pages 2088–2101, Swansea UK, 1988. Pineridge Press, C. Taylor, W.G. Habashi. and M. M. Hafez. eds.

[68] P. L. Roe. A survey of upwind differencing techniques. In *Lecture notes in Physics 323, 11th International Conference on Numerical Methods in Fluid Dynamics.* D. L. Dwoyer, M. Y. Hussaini, R. G. Voigt, Eds., Springer-Verlage, 1989.

[69] P. L. Roe. Sonic flux formulae. *SIAM Journal on Scientific and Statistical Computing*, 13(2):611–630, 1992.

[70] R. Sacks-Davis. Error estimates for a stiff differential equation procedures. *Mathematics of Computation*, 31:939–953, 1977.

[71] L. F. Shampine. Type-insensitive ODE codes based on implicitA.stable formulas. *Mathematics of Computation*, 36:499–510, 1981.

[72] L. F. Shampine. Type insensitive O.D.E codes based on implicit A(α) stable fo rmulas. *Mathematics of Computation*, 39:109–123, 1982.

[73] L. F. Shampine. Global error estimation for stiff ODEs. In *Proc. Dundee Conference on numerical Analysis, Lecture Notes in Mathematics, 1066.* Springer-Verlag, New York, 1984.

[74] L. F. Shampine and M. W. Reichelt. The Matlab ODE suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.

[75] C.-W. Shu and S. Osher. Efficient implementation of essential non-oscillatory shock-capturing schemes II. *Journal of Computational Physics*, 83:32–78, 1989.

[76] J.-S. Shuen, M.-S. Liou, and B. van Leer. Inviscid flux-splitting algorithm for real gases with non-equilibrium chemistry. *Journal of Computational Physics*, 90:371–395, 1990.

[77] D. R. Stall and H. Prophet. *JANAF Thermochemical Tables.* National Standard Reference Data Series, 1971.

[78] D. F. Griffiths A.M. Stuart and H.C. Yee. Numerical wave propagation in an advection equation with a nonlinear source term. *SIAM Journal on Numerical Analysis*, 29:1244–1260, 1992.

[79] A. Suresh and M.-S.Liou. The Osher scheme for non-equilibrium reacting flows. *International Journal for Numerical Methods in Fluids*, 15:219–232., 1992.

[80] P. K. Sweby. "TVD" schemes for inhomogeneous conservation laws. In *Notes on Numerical Fluid Mechanics Volume 24, Non-linear Hyperbolic Equations-Theory, Computation Methods and Application.* J. Ballmann, R. Jeltsch, Eds.,Verlage pp. 599-607, 1989.

[81] T. Tang. Convergence anlaysis for operator splitting method applied to conservation laws with stiff source terms. *SIAM Journal on Numerical Analysis*, 35(5):1939–1968, October 1998.

[82] A. Tomlin, M. Berzins, J. M. Ware, J. Smith, and M.J. Pilling. On the use of adaptive gridings methods for the modelling chemical transport from multiscale sources. *Atmospheric Environment*, 31(18):2945–2959, 1997.

[83] V. T. Ton, A. R. Karagozian, F. E. Marble, S. J. Osher, and B. E. Engquist. Numerical simulation of hight speed chemically reacting flow. *Theoret. Comput. Fluid Dynamics*, 6:161–179, 1994.

[84] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer-Verlag, Berlin Heidelberg New York, 1997.

[85] B. van Leer. Towards the ultimate conservative difference scheme. V. a second order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[86] B. van Leer. On the relation between the upwind differencing schemes of Godunov, Engquist-Osher and Roe. *SIAM Journal on Scientific and Statistical Computing*, 5:1–20, 1984.

[87] B. van Leer, J. L. Thomas, P. L. Roe, and R.W. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equation. In *Proc. AIAA 8th Computational Fluid Dynamics Conference*, pages 36–41. AIAA Paper 87-1104, 1987.

[88] M. van Loon. *Numerical Methods in Smog Prediction.* PhD thesis, CWI, Amsderdam, 1996.

[89] J. Varah. Stability restrictions on second-order, three level finite difference sche mes for parabolic equations. *SIAM Journal on Numerical Analysis*, 17:300–309, 1980.

[90] J.G. Verwer. Guass-Seidel iteration for stiff odes from chemical kinetics. *SIAM Journal on Scientific Computing*, 15:1243–1250, 1994.

[91] N. P. Waterson and H. Deconinck. A unified approach to the design and application of bounded higher-order convection schemes. In *Proceedings of the 9th International Conference on Numerical Methods in Laminar and Turbulent Flow*, pages 881–892. Pineridge Press, 1995.

[92] R. P. Wayne. *Chemistry of Atmosphere.* Clarendon Press,Oxford, second edition, 1991.

[93] H.C. Yee. *A class of high-resolution explicit and implicit shock capturing methods.* Von Karman Institute Belgium., 1987.

[94] Z. Zlatev, J. Christensen, and O. Hov. An Eulerian air polution model for Europe with non-linear chemistry. *Journal of Atmospheric Chemistry*, 15:1–37, 1992.