To the Graduate Council:

I am submitting herewith a thesis written by Ramanan Sankaran entitled "Assessment of numerical issues in one-dimensional detonation wave representation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

Charles L. Merkle, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

Assessment of Numerical Issues in One-Dimensional Detonation Wave Representation

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Ramanan Sankaran

August 2000

Acknowledgments

I am greatly indebted to my major professor Dr Charles Merkle for his guidance, encouragement and support in my academics and research work He provided both the freedom to pursue my ideas and the experience and knowledge to help keep my work grounded and in context I am grateful to my research group members Ding Li, Rajendran Mohanraj, Guoping Xia and our group secretary Brenda Brown for all their support during my stay at UTSI

That this two year period will definitely be one of the nicest times in my life is owed to my friends here, with whom i would love to spend many more years together Ashok (Baasha of UTSI) was an understanding and caring friend The time I spent here at UTSI in the company of Ashok and Mehul will remain fresh in my memories for a long time This being my first stay away from my family made your friendship even more important And then there is Swapnil, Chandra, Abraham and Subha I am going to miss our dinner (talkathon) meetings Studies and work has never been more fun

Abstract

This thesis was an attempt to simulate the detonation waves and contact surfaces in unsteady flows with improved accuracy and efficiency through the use of adaptive mesh refinement (AMR) In the present work, the problem dealt with is the simulation of the working cycles of a pulsed detonation engine (PDE) A flexible code which can be used for any unsteady flow simulation was developed The analysis was based on the quasi one-dimensional Euler equations and the reaction rate was modelled using a one-step irreversible reaction equation The numerical simulations were carried out using two numerical schemes, namely, Roe's approximate Riemann solver and the advection upstream splitting method (AUSM)

Results of this numerical study show the importance and effects of increasing the spatial resolution The use of adaptive mesh refinement made it possible to increase the spatial resolution with insignificant increases in cost of computations The results also show that the contact surfaces cannot be captured accurately merely by increasing the spatial resolution, due to the high innate numerical diffusion of the flux schemes The possibility of confining an interface to a few cell distances by adding a suitable confinement term has also been discussed

Contents

1 Introduction 1

2 Formulation of Numerical Schemes 4

2 1 Conservation Equations 4

2 2 Finite Volume Formulation . . 6

2 3 Approximate Riemann Solvers . 6

2 4 Roe's Approximate Riemann Solver 8

2 5 The Advection Upstream Splitting Method or AUSM$^+$ 10

2.5 1 Definition of $a_{1/2}$ . . 11

2 5 2 Mach Number and Pressure Functions 12

2 5 3 Numerical Fluxes . . 13

3 Adaptive Mesh Refinement 15

3 1 FTT Structure and Implementation 16

3 2 Integration Procedure 19

3 3 Mesh Refinement . 22

4 Pulsed Detonation Engine Model 25

4 1 PDE operational stages . . . 25

4 2 Computational Model 27

4 3 Boundary conditions 28

5 Results and Discussion 31

5 1 Detonation Initiation and Mesh Size 31

5 2 Contact Discontinuity 37

5 3 Adaptive Mesh Refined Simulations 49

5 4 Improved methods for resolving a contact discontinuity 64

5 4 1 Confinement of a passive scalar discontinuity 67

5 4 2 Confinement of contact discontinuities in vector systems . . 69

6 Conclusions and Recommendations 75

Bibliography 77

Appendix 81

A Jacobians and Eigensystem for Reactive Euler Equations 82

Vita 85

List of Figures

2 1 A typical quasi-one-dimensional mesh . 7

3 1 Logical relationship between cells of a one-dimensional, binary, fully threaded tree . . 17

3 2 Relationship between cells and octs in a fully threaded tree . 18

4 1 Various stages in the operation of a pulsed detonation engine . 26

5 1 Pressure and Temperature profiles at $t = 1\ 0 \times 10^{-5} s$, for a uniform mesh of 1024 cells 32

5 2 Pressure and Temperature profiles at $t = 1\ 0 \times 10^{-5} s$, for a uniform mesh of 2048 cells . . 33

5 3 Location of Reaction front as a function of time for various mesh sizes during the initiation phase 35

5 4 Location of Reaction front as a function of time for various mesh sizes on a longer time scale 36

5 5 Comparison of peak pressure for various mesh sizes during the initiation phase . 38

5 6 Comparison of peak pressure for various mesh sizes on a longer time scale. 39

5 7 Comparison of peak temp for various mesh sizes during the initiation phase 40

5 8 Comparison of peak temp for various mesh sizes on a longer time scale 41

5 9 $x-t$ diagram during the initiation period for a uniform mesh of 1024 cells 42

5 10 $x-t$ diagram during the initiation period for a uniform mesh of 2048 cells 43

5 11 Density (ρ) and Reactant Mass fraction (Z) profiles at time $t = 1\ 8\times10^{-4}$s, for different mesh sizes 45

5 12 Plot of thickness of interface vs mesh size 46

5 13 Comparison of Density (ρ) and Reactant Mass Fraction (Z) profiles obtained using different numerical schemes, for a mesh of 1024 cells . 48

5 14 Pressure plot and mesh structure at three different instants of time during the propagation of the detonation wave 50

5 15 Location of reaction front as a function of time for a uniform grid and AMR grid . . . 51

5 16 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\ 3 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\ 1$ 53

5 17 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\ 6 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\ 1$ 54

5 18 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\ 0 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\ 1$ 55

5 19 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\ 3 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\ 05$ . . 57

5 20 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,6 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\,05$ 58

5 21 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\,0 \times 10^{-4}s$, during the filling phase, with $\varepsilon^c = 0\,05$ 59

5 22 Plots of density and reactant mass fraction at time $t = 2\,0 \times 10^{-4}s$, during the filling phase for a uniform grid and AMR grid . . 60

5 23 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,3 \times 10^{-4}s$, during the filling phase, using a tracking variable for the interface . 61

5 24 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,6 \times 10^{-4}s$, during the filling phase, using a tracking variable for the interface . . . 62

5 25 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\,0 \times 10^{-4}s$, during the filling phase, using a tracking variable for the interface . 63

5 26 Comparison of density and reactant mass fraction profiles, with and without tracking at time $t = 1\,6 \times 10^{-4}s$ 65

5 27 Comparison of density and reactant mass fraction profiles, with and without tracking at time $t = 2\,0 \times 10^{-4}s$ . . 66

5 28 Advection of two contact surfaces with Confinement 72

5 29 Advection of two Z-interfaces with Confinement 73

Nomenclature

a	speed of sound
A	Jacobian of flux vector E with respect to conservative variables Q
$\mathcal{A}$	area of cross section of tube
C	reactant
C_p	specific heat at constant pressure
D	product
e	specific total energy
E	flux vector
E_a	activation energy
h	specific stagnation enthalpy
H	source term vector
K	pre-exponential factor
l	level of the tree
L	length of computational domain
m	mass flux
M	Mach number
$\mathcal{M}$	Mach number polynomial
MW	molecular weight
$\widehat{M}$	matrix of right eigenvectors of flux Jacobian A
p	pressure
$\mathcal{P}$	pressure polynomial

q_0	heat release per unit mass
Q	vector of conservative variables
$\widehat{Q}$	vector of characteristic variables
R	specific gas constant
R_u	universal gas constant
s	entropy
t	time
T	temperature
u	velocity
v_D	detonation velocity
V	volume of a cell
x	cartesian coordinate
Z	reactant mass fraction
γ	ratio of specific heats
ϵ	specific internal energy
ε	confinement parameter
λ	eigenvalues of flux Jacobian A
Λ	diagonal matrix of eigenvalues λ
ξ	refinement indicator
ρ	density
σ	CFL number

Subscripts

$1/2$	interface quantity
i	mesh point location in x direction
L, R	left and right states

Chapter 1

Introduction

There are abundant CFD applications, which involve the simulation of shocks and other discontinuities A core objective of many CFD algorithms is to simulate these features with high accuracy and minimal cost of computation The present work deals with one such application, the simulation of a pulsed detonation engine (PDE), in which there is a need to accurately simulate a detonation wave and contact surface The PDE is a novel alternative to conventional gas turbine or rocket engines that can provide improvements in performance or costs for space propulsion Some anticipated advantages of PDEs are (a) compactness, due to the high energy densities involved, (b) lower specific fuel consumption due to higher thermodynamic efficiency compared to a constant pressure combustion process and (c) reduction in mechanical complexity due to the lack of compressor and turbine The PDE model used and all simulations carried out as part of this work were based on a recent study by Mohanraj [10] and is aimed at generating a flexible code and improving the efficiency of computation.

The operation of a PDE can be divided into three phases (a) the *fill time* during which the chamber is filled with reactants, (b) the *detonation travel time* during which

the detonation is initiated and traverses through the chamber burning up the reactants and (c) the *blowdown time* during which the bulk of the combustion products are expelled by a system of compression and rarefaction waves After the blowdown process the fill process starts again and the cyclic operation continues A detailed description of the PDE operation is presented in chapter 4 (see also figure 4 1) A finely refined mesh is required for accurately simulating the detonation waves and contact surfaces However during some parts of the cycle, like the blowdown process, no sharp flow features are present and the computation could as well be carried out using a coarse mesh In such a situation, one attractive way to solve a problem is to use adaptive mesh refinement (AMR)

Adaptive mesh refinement is used to increase the spatial and temporal resolution of the numerical simulation while minimizing the computational cost The mesh is not fixed in time and is constantly evolving to match the instantaneous requirements AMR makes it possible to selectively refine the mesh only in some regions of the computational domain where accuracy is of significance, while retaining a coarse mesh where it is not This helps in minimizing the cost of computation

In the present work, operation of a PDE has been modelled using an AMR simulation The analysis is based on the quasi one-dimensional Euler equations using a single progress variable to model the reactants and products The numerical simulations were carried out using two numerical schemes, namely, Roe's approximate Riemann Solver [13] and AUSM$^+$ [8] Both the solutions are first order accurate in space and time The procedures required for the AMR simulation were coded using the Fully Threaded Tree (FTT) data

structure of Khokhlov [6] Finally an attempt was made to effectively capture the contact surfaces by adding artificial dissipation that has the effect of confining the spreading contact surfaces

Chapter 2

Formulation of Numerical Schemes

The present chapter lays the foundation for the numerical simulation of the detonation phenomenon. The reactive Euler equations for the system are expressed in the conservation form and then two different numerical schemes are formulated for those equations

2.1 Conservation Equations

The quasi-one-dimensional reactive Euler equations where a single reactant C is converted to a single product D by a one-step irreversible chemical reaction governed by Arrhenius kinetics, may be written in conservative form as

$$\frac{\partial QA}{\partial t} + \frac{\partial EA}{\partial x} = HA \tag{2.1}$$

where $\mathcal{A} = \mathcal{A}(x)$ is the local cross section area, Q is the vector of conservative variables, E is the flux vector and H is the source term

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho e \\ \rho Z \end{pmatrix}, \quad E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (\rho e + p)\, u \\ \rho u Z \end{pmatrix}, \quad H = \begin{pmatrix} 0 \\ \frac{p}{\mathcal{A}} \left(\frac{d\mathcal{A}}{dx} \right) \\ 0 \\ -K \rho Z \exp^{-E_a / R_u T} \end{pmatrix} \tag{2 2}$$

Here ρ, u, p, T, e, Z, E_a and R_u are the density, velocity, pressure, temperature, specific total energy, reactant mass fraction, activation energy and universal gas constant respectively K is a constant termed as the pre-exponential factor It is an empirical parameter that determines the reaction rate The following equations are used to close the system 2 1

$$\begin{aligned} e &= \epsilon + \frac{1}{2} u^2 + q_0 Z \\ p &= p(\rho, T) \end{aligned} \tag{2 3}$$

Here ϵ is the specific internal energy (in J/kg) and q_0 is the heat release parameter (in J/kg) for the chemical reaction C $\rightarrow$ D It is assumed that the molecular weights and the specific heats are constants for the reactant and product The various Jacobians and matrices for this system of equations, essential for the development of flux schemes, are listed in Appendix A

2.2 Finite Volume Formulation

Let a domain of length L be split into N cells A cell of index i, where $i \in [1, N]$, is characterized by its centroid x_i, volume V_i, and location of its left and right faces at $x_{i-1/2}$ and $x_{i+1/2}$ respectively The variation of area along the length is given by $\mathcal{A} = \mathcal{A}(x)$ Given this description of a cell, equation 2 1 is integrated over the volume of a cell i in the usual fashion This results in an explicit update equation for the unknown vector Q ,

$$\Delta Q_i = -\frac{\Delta t}{V_i}\left[\widetilde{E}_{i+1/2}\mathcal{A}_{i+1/2} - \widetilde{E}_{i-1/2}\mathcal{A}_{i-1/2}\right\} + \Delta t\, H_i \tag{2 4}$$

$$Q_i^{n+1} = Q_i^n + \Delta Q_i \tag{2 5}$$

Here $\widetilde{E}$ is the numerical flux computed on the faces of the cells explicitly using one of the numerical schemes to be discussed below In the finite-volume formulation 2 4, the differences among all the numerical schemes lie essentially in the definition of the numerical flux $\widetilde{E}_{i+1/2}$ evaluated at the cell interface Figure 2 1 shows a typical quasi-one-dimensional mesh along with its cells, their centroid and faces Two different upwind numerical schemes for computing the numerical flux are presented in this chapter

2.3 Approximate Riemann Solvers

One way of designing an upwind numerical scheme is through the use of the so-called Riemann solvers or approximate Riemann solvers In these methods the solution is considered as piecewise constant over each mesh cell at a fixed time and the evolution

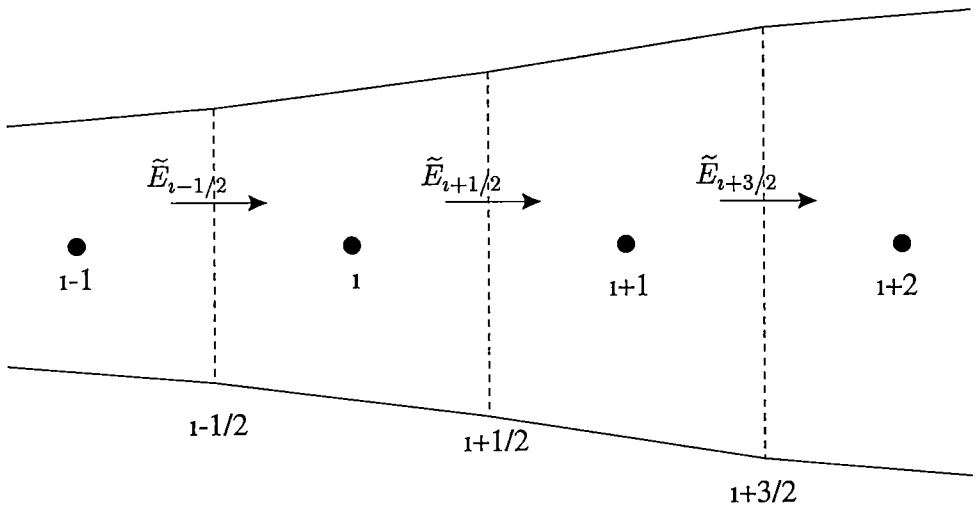


Figure 2 1· A typical quasi-one-dimensional mesh

of the flow to the next time step results from the wave interactions originating at the boundaries between adjacent cells The cell interfaces separate two different fluid states Q_L at the left side and Q_R at the right side and the resulting local interaction can be exactly resolved since the initial conditions at time $t = n\Delta t$ correspond to the Riemann or shock tube problem This problem will have an exact solution composed of a shock wave, a contact discontinuity and an expansion fan separating regions of uniform flow conditions Each wave carries information in an upwind manner and hence the resulting state will only depend on the local physical properties The new piecewise constant approximation at time $t = (n + 1)\Delta t$ is then obtained by averaging over each cell, the fluid states resulting from the perturbation waves This produces an explicit conservative scheme of first-order accuracy

Since the exact solution of the Riemann problem requires the resolution of a non-linear algebraic equation which can be quite time consuming, approximate Riemann solutions are frequently considered to reduce the computational work at each interface

Two interesting Riemann solvers developed by Roe [13] and Liou [8] will be summarized in the following sections The flux expressions of these numerical schemes are expressed in a form suitable for comparison

2.4 Roe's Approximate Riemann Solver

The approximate Riemann solver developed by Roe [13] is based on a characteristic decomposition of the flux difference while ensuring the conservation properties of the scheme Roe constructs a mean value Jacobian $\bar{A}$ such that the following conditions are met

1 $E(Q_R) - E(Q_L) = \bar{A}(Q_L, Q_R)(Q_R - Q_L)$

2 The matrix $\bar{A}(Q_L, Q_R)$ has real eigenvalues and is diagonalizable (has linearly independent eigenvectors) (See appendix A)

3 $\bar{A}(Q, Q) = A(Q) = \partial E/\partial Q$, so that consistency is satisfied

As the first step an averaged state $\bar{Q} = \bar{Q}(Q_L, Q_R)$, known as the Roe average is defined For the system 2 1, the Roe average is defined by the following relationships

$$\begin{aligned} \bar{\rho} &= \sqrt{\rho_L \rho_R} \\ \bar{u} &= \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \bar{h} &= \frac{h_L\sqrt{\rho_L} + h_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \bar{Z} &= \frac{Z_L\sqrt{\rho_L} + Z_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \end{aligned} \tag{2 6}$$

where $h = e + p/\rho$ Further, the flux E in equation 2.2 can be expressed as a sum of the convective and pressure terms

$$E = E^c + P \tag{2 7}$$

$$E^c = m\Psi, \quad m = \rho M a, \quad \Psi = \begin{pmatrix} 1 \\ u \\ h \\ Z \end{pmatrix}, \quad P = \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \end{pmatrix} \tag{2 8}$$

Here m, M and a are the mass flux, Mach number and speed of sound respectively Using these definitions the numerical flux for the first order Roe's scheme $\widetilde{E}_{i+1/2} = \widetilde{E}\left(Q_i, Q_{i+1}\right)$ on the cell face straddling the ith and $i+1$th cells can be expressed as,

$$\widetilde{E}_{i+1/2} = \frac{1}{2}\left((m\Psi)_i + (m\Psi)_{i+1}\right) - \frac{1}{2}|\bar{A}|_{i+1/2}\left(Q_{i+1} - Q_i\right) + \frac{1}{2}\left(P_i + P_{i+1}\right) \tag{2 9}$$

$$|\bar{A}|_{i+1/2} = \widehat{M}_{i+1/2}|\Lambda|_{i+1/2}\widehat{M}^{-1}_{i+1/2} \tag{2 10}$$

Here the matrix $|A|$ can be referred to as "the magnitude of A" It is a positive definite matrix and its eigenvalues are given by the magnitudes of the eigenvalues of A The matrix $\widehat{M}$ is composed of the right (or column) eigenvectors of A and $|\Lambda|$ is the diagonal matrix formed from the corresponding absolute values of the eigenvalues All the quantities in equation 2 10 are evaluated using the average value $\bar{Q} = \bar{Q}\left(Q_i, Q_{i+1}\right)$ as defined in 2 6

One disadvantage of Roe's approximate Riemann solver is that it only enforces jump conditions across each cell and does not enforce a condition on entropy So both expan-

sion shocks and compression shocks are allowed In the present notation, the Rankine-Hugoniot jump conditions are identically given by $E_{i+1} = E_i$ The entropy condition that $s_{i+1} \geq s_i$ (or conversely, if the shock is propagating in the opposite direction) is not contained in the relation $E_{i+1} = E_i$ and must be added externally Harten [4] advocates a technique in which a local expansion fan is introduced in the approximate Riemann solution when an expansion is detected through a sonic point This is realized by limiting any vanishing eigenvalue to a small value $\bar{\epsilon}$, which rules out the possibility of an expansion shock As described by Harten [4], the use of this value $\bar{\epsilon}$, introduces an intermediate state (between the states Q_L and Q_R) that simulates the diffusion present in an exact Riemann solver For cases where an eigenvalue λ changes sign, the modified eigenvalue is defined as

$$|\bar{\lambda}| = \begin{cases} |\bar{\lambda}|_{i+1/2} & \text{if } |\bar{\lambda}|_{i+1/2} \geq \bar{\epsilon} \\ \bar{\epsilon} & \text{if } |\bar{\lambda}|_{i+1/2} < \bar{\epsilon} \end{cases} \tag{2 11}$$

The quantity $\bar{\epsilon}$ is obtained from,

$$\bar{\epsilon} = \max\left[0, \left(\bar{\lambda}_{i+1/2} - \lambda_i\right), \left(\lambda_{i+1} - \bar{\lambda}_{i+1/2}\right)\right] \tag{2 12}$$

2.5 The Advection Upstream Splitting Method or AUSM$^+$

The AUSM$^+$ scheme of Liou [7, 8, 9] is formulated by basically distinguishing between the convection and acoustic waves and treating them as two physically distinct processes The convective terms are upstream-biased using an appropriately defined cell-interface

velocity, while the pressure term is strictly dealt with by using acoustic waves This explains the scheme's name, advection upstream splitting method Following the procedure used for the formulation of Roe's method, the flux E is expressed as,

$$E = E^c + P \tag{2 13}$$

$$E^c = m\Psi, \quad \dot{m} = \rho M a, \quad \Psi = \begin{pmatrix} 1 \\ u \\ h \\ Z \end{pmatrix}, \quad P = \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \end{pmatrix} \tag{2 14}$$

The AUSM$^+$ flux is defined in terms of the interface mass flux $\dot{m}_{1/2}$, which is expressed in terms of the Mach number at interface $M_{1/2}$, and speed of sound $a_{1/2}$ Note that the interface quantities are denoted by the subscript $1/2$

2.5.1 Definition of $a_{1/2}$

It is necessary to use a common speed of sound $a_{1/2}$ rather than a_i or a_{i+1} to achieve a unification of the splittings of both Mach number M and unknown vector Q [7, 8] For the present unsteady flow test case, the judicious choice of an interface speed of sound will be,

$$a_{1/2} = \frac{1}{2}\left(a_i + a_{i+1}\right) \tag{2 15}$$

Utilizing this common speed of sound, the mach numbers in the adjoining cells may be defined as

$$M_i = \frac{u_i}{a_{1/2}}, \quad M_{i+1} = \frac{u_{i+1}}{a_{1/2}} \tag{2 16}$$

2.5.2 Mach Number and Pressure Functions

The interface mass flux and pressure are defined in terms of certain polynomials of the Mach number M These polynomials are called the Mach number and pressure functions The following Mach number and pressure functions are defined to facilitate further analysis Let,

$$\mathcal{M}^{\pm}_{(1)}(M) = \frac{1}{2}(M \pm |M|) \tag{2 17}$$

$$\mathcal{M}^{\pm}_{(2)}(M) = \begin{cases} \mathcal{M}^{\pm}_{(1)}(M), & \text{if } |M| \geq 1, \\ \pm\frac{1}{4}(M \pm 1)^2, & \text{otherwise} \end{cases} \tag{2 18}$$

$$\mathcal{M}^{\pm}_{(4,\beta)}(M) = \begin{cases} \mathcal{M}^{\pm}_{(1)}(M), & \text{if } |M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}(M)\left[1 \mp 16\beta\mathcal{M}^{\mp}_{(2)}(M)\right], & \text{otherwise} \end{cases} \tag{2.19}$$

These polynomials were devised to meet certain properties, such as consistency, continuous differentiability, symmetry and monotonicity; see [7] for details They are the basic ingredients used for defining the interface mass flux $m_{1/2}$ Let the pressure functions be defined by,

$$\mathcal{P}^{\pm}_{(3)} = \begin{cases} \mathcal{M}^{\pm}_{(1)}(M)/M, & \text{if } |M| \geq 1, \\ \pm\mathcal{M}^{\pm}_{(2)}(M)(2 \mp M), & \text{otherwise} \end{cases} \tag{2 20}$$

$$\mathcal{P}^{\pm}_{(5,\alpha)}(M) = \begin{cases} \mathcal{M}^{\pm}_{(1)}(M)/M, & \text{if } |M| \geq 1, \\ \pm\mathcal{M}^{\pm}_{(2)}(M)\left[(2 \mp M) - 16\alpha M \mathcal{M}^{\mp}_{(2)}(M)\right], & \text{otherwise} \end{cases} \tag{2 21}$$

The numerals in the subscript of $\mathcal{M}$ and $\mathcal{P}$ indicate the degree of the polynomial used in the range $|M| < 1$ For $|M| > 1$ all polynomials are first order (α, β) are parameters as indicated Considering a few more conditions on the polynomials (see [9]), yields the following values for α and β,

$$\alpha = \frac{3}{16} \text{ and } \beta = \frac{1}{8}$$

2.5.3 Numerical Fluxes

In the AUSM$^+$ scheme, the interface mass flux is defined using the Mach number functions as follows (This is to ensure that the flux splitting will also satisfy the properties of consistency, monotonicity, etc)

$$M_{1/2}(M_i, M_{i+1}) = \mathcal{M}^{+}_{(4,\beta)}(M_i) + \mathcal{M}^{-}_{(4,\beta)}(M_{i+1}) \tag{2 22}$$

$$m_{1/2} = \begin{cases} \rho_L M_{1/2} a_{1/2}, & \text{if } M_{1/2} \geq 0 \\ \rho_R M_{1/2} a_{1/2}, & \text{otherwise} \end{cases} \tag{2 23}$$

Next the interface pressure is defined as,

$$p_{1/2}(Q_i, Q_{i+1}) = \mathcal{P}^{+}_{(5,\alpha)}(M_i)\, p_i + \mathcal{P}^{-}_{(5,\alpha)}(M_{i+1})\, p_{i+1} \tag{2 24}$$

Then the full numerical flux, $\widetilde{E}_{i+1/2}$ is

$$\widetilde{E}_{i+1/2} = \frac{1}{2}\left[m_{1/2}\left(\Psi_i + \Psi_{i+1}\right) - |m_{1/2}|\left(\Psi_{i+1} - \Psi_i\right)\right] + p_{1/2} \tag{2 25}$$

Comparing the AUSM$^+$ flux equation 2 25 with Roe's flux expression 2 9, the following observations can be made

1 The first term on the RHS of equation 2 25 is clearly not a simple average of the flux E at the i and $i+1$ states, but rather a Mach number weighted average

2 The major difference between the two schemes is the dissipation term In equation 2 25, the dissipation coefficient $|m_{1/2}|$ is merely a scalar, while in Roe's scheme 2 9, the dissipation coefficient is the matrix $|\bar{A}|_{i+1/2}$

3 The AUSM$^+$ scheme does not involve computation of the Jacobian matrix, and it always involves only the common term $m_{1/2}$ for any additional conservation laws This reduces the computational cost, and again, the cost is only linearly increased when any additional conservation equations are considered As will be seen later, the AUSM$^+$ scheme performs as well as Roe's scheme and costs less

Chapter 3

Adaptive Mesh Refinement

Adaptive Mesh Refinement (AMR) is used to increase the spatial and temporal resolution of a numerical simulation with insignificant increases in cost of computation In the present unsteady flow simulation, the refined portion of the mesh has to closely follow those regions of the flow where gradients are large (shocks, detonation waves and contact surfaces), to prevent them from passing out of the fine mesh region as the iteration proceeds This necessitates periodic rebuilding of the mesh The individual computational cells can be organized on a tree structure, so that each cell can be refined or unrefined separately from the others, as needed While this tree structure is natural for mesh refinement, it requires additional features to allow an efficient data structure to be declared The Fully-Threaded Tree (FTT) algorithm of Khokhlov [6] satisfies these requirements In the FTT structure every cell has an easy access to its children, neighbors and parents It could be said that an FTT is a tree threaded in all possible directions, which explains the reason behind its name This tree structure allows all operations and modifications of the tree to be performed in parallel, which is quite promising for use with parallel computers The tree structure and implementation details of FTT as

applied by Khokhlov [6] are explained here

3.1 FTT Structure and Implementation

The computational domain of length L, is subdivided into a number of cells of various sizes $1/2, 1/4, 1/8,$ of L Cells are logically organized in a binary tree with the entire computational domain being the root With every cell i the following information can be associated

iLv(i) — level of the cell in the tree

iKy(i) — TRUE/FALSE if cell is split/unsplit

iPr(i) — pointer to a parent cell

iCh(i,j) — pointers to children, j=1,2

iNb(i,j) — pointers to neighbors, j=1,2

The cell size is related to the level *iLv(i)* by $\Delta_i = L/2^{iLv(i)}$ Cells in the tree are either split (have children) or leaves (do not have children) Logical relations between cells in the tree, and the directions of various pointers are illustrated in figure 3 1 for the one-dimensional binary tree In figure 3 1, the root (cell 1) represents the entire computational domain It has two children (cells 2 and 3), each representing half of the domain Cell 2 is further subdivided into two cells (cells 4 and 5) *Neighboring leaves are not allowed to differ in size by more than a factor of 2* The neighbor-neighbor relation is not reciprocal for leaves of different sizes that face each other In Fig 3 1, cell

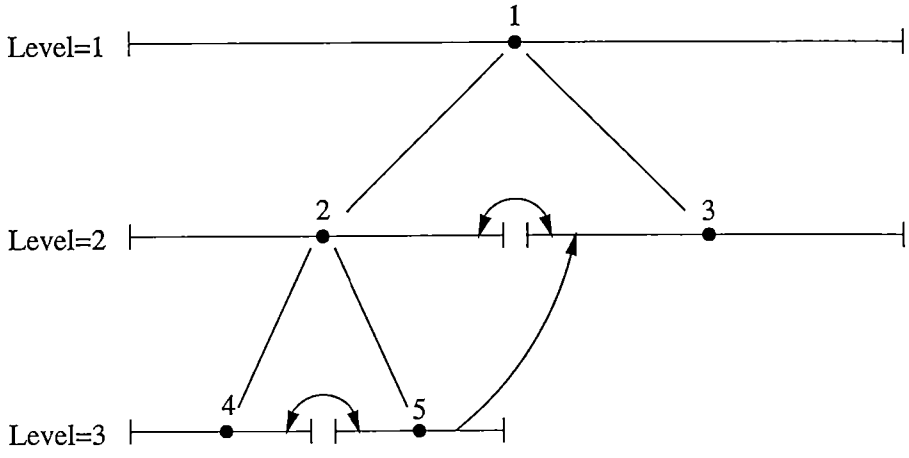


Figure 3 1 Logical relationship between cells of a one-dimensional, binary, fully threaded tree

5 has cell 3 as its neighbor, but cell 3 has cell 2 as its neighbor and not cell 5 A jth neighbor of a cell i either has the same size as the cell itself, $\Delta_{iNb(i,j)} = \Delta_i$, or it is two times larger, $\Delta_{iNb(i,j)} = 2\Delta_i$ In the former case, the neighbor may be a leaf or a split cell In the latter case, it can only be a leaf

In the FTT structure, all cells are organized in groups called "octs" Each oct contains two cells (When extended to multi-dimensions, every oct contains 4 cells in a two-dimensional configuration and 8 cells in three-dimensions). The name "oct" is chosen because it contains 8 cells in three-dimensions The FTT structure implemented is illustrated in figure 3 2 Each cell stores the physical state vector Q^n and ΔQ associated with it It also has a pointer to an oct which contains its children, if any, or a *nil* pointer Each octet knows its level, *OctLv*, which is equal to the level of the oct's cells Each oct has a pointer *OctPr* to a parent cell It also has pointers *OctNb(k)* to parent cells of neighboring octs The number of neighbors for 1, 2 and 3 dimensions are k=2, 4 and 6 respectively Octs also contain the coordinates $r(x, y, z)$ of their centers, which is also

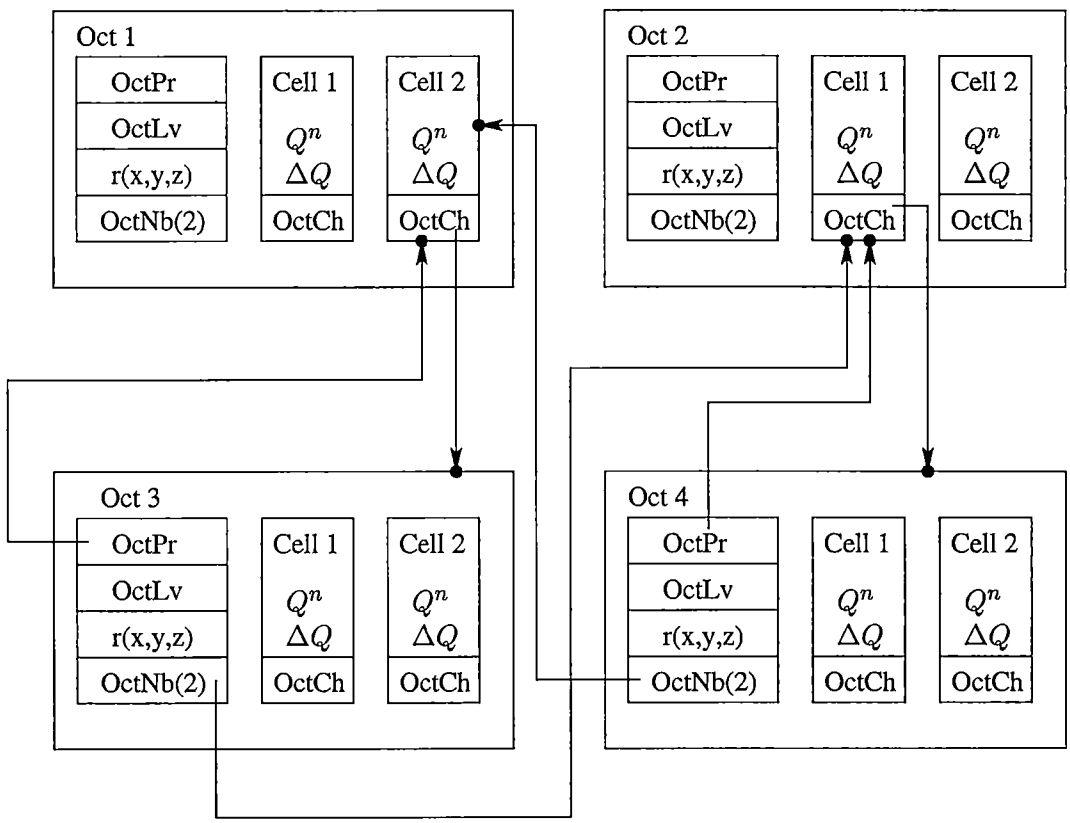


Figure 3 2 Relationship between cells and octs in a fully threaded tree Pointers from octs to cells and from cells to octs are indicated by arrows

the centroid of their parent cells For the uniform area case, the coordinates of cells that belong to an oct can be found by adding or subtracting $\Delta_i/2$ from the corresponding oct's coordinates But for the variable area cases, the location of the centroid needs to be computed Hence, it is economical to store the coordinates rather than compute them every time they are needed

3.2 Integration Procedure

Integration in time on the tree can utilize the usual flux evaluation algorithms used for grids, but requires a different time-stepping strategy to be computationally efficient This strategy is quite different from that typically used on regular grids In the integration procedure described below, integration in time and tree refinement are coupled together and time stepping at different levels of the tree and tree refinements of these levels are interleaved Also the computations are organized not on a cell-by-cell, but on a face-by-face basis At every face, fluxes are evaluated, changes to cell values on the left and on the right of the face are applied, and then fluxes are discarded

The finite volume formulation 2 4 has to be used to update Q with different time steps $\Delta t(l)$ at different levels of the tree l, $l_{min} \leq l \leq l_{max}$, where l_{min} and l_{max} are the minimum and maximum levels of leaves (l_{min} represents the cells on the coarsest grid and l_{max} those on the finest grid) A global time step, on the coarsest grid, $\Delta t = \Delta t(l_{min})$ is determined from the CFL condition

$$\Delta t = \sigma \frac{2^{-l_{min}} L}{\max_i (a_i + |u_i|)} \tag{3 1}$$

where a is the sound speed, $\sigma < 1$ is the CFL number, and the maximum in equation 3.1 is taken over all leaves i in all the levels. Time steps at various levels are scaled back proportionate to cell size,

$$\Delta t(l) = 2^{l_{min}-l}\Delta t \tag{3.2}$$

Integration at different levels of the tree is interleaved with tree refinement. Let the procedure of advancing level l one step $\Delta t(l)$ in time be designated as $\mathcal{A}(l)$, and the procedure of tree refinement at level l as $\mathcal{R}(l)$. The $\mathcal{R}$ procedure consists of refining leaves of level l and unrefining split cells of level l according to certain refinement criteria. This procedure is described in section 3.3. The advancement procedure $\mathcal{A}$ is described below.

Let the right and left neighbors of cell i be $i+$ and $i-$, respectively. The state vector at the beginning and end of a global time step is Q^n and Q^{n+1} respectively. The $\mathcal{A}(l)$ procedure is described in the form of the following pseudocode:

```
for (leaves i of level l-1){ if(i has split neighbors) ΔQ_i = 0
for (leaves i of level l){
    if (i+ is a leaf or boundary){
       Compute fluxes Ẽ_{i+1/2} at the (i,i+) face;
       ΔQ_i = ΔQ_i − Ẽ_{i+1/2} A_{i+1/2} + H_i V_i
       ΔQ_{i+} = ΔQ_{i+} + Ẽ_{i+1/2} A_{i+1/2}
    }
    if (i− is a leaf of level l−1 or boundary){
```

Compute fluxes $\widetilde{E}_{i-1/2}$ at the $(i, i-)$ face;

$\Delta Q_i = \Delta Q_i + \widetilde{E}_{i-1/2}\mathcal{A}_{i-1/2}$

$\Delta Q_{i-} = \Delta Q_{i-} - \widetilde{E}_{i-1/2}\mathcal{A}_{i-1/2}$

}

}

for (cells i of level l){

if (i is a leaf)

$Q_i^n = Q_i^n + \Delta Q_i * \Delta t(l)/V_i$

$\Delta Q_i = 0$

else

$$Q_i^n = \frac{1}{2}\sum_{j=1}^{2} Q_{iCh(i,j)}^n$$

}

The global time step of integration consists of going through all levels of the tree, starting with l_{min}, and performing a sequence of refinements and advances at every tree level Advancing the solution by one global time step is done through a call for the procedure $\mathcal{S}(l)$, where the procedure $\mathcal{S}(l)$ is a combination of advancing and refinement procedures,

$$\mathcal{S}(l) = \mathcal{R}(l)\mathcal{S}(l+1)\mathcal{A}(l)\mathcal{S}(l+1)\mathcal{A}(l) \tag{3 3}$$

and $\mathcal{S}(l)$ does nothing if $l > l_{max}$ All procedures in 3 3 are performed from left to right For example, for $l_{min} = 6$ and $l_{max} = 8$, the sequence generated by 3 3 would be

$\mathcal{S}(6) = [\mathcal{R}(6)\ [\mathcal{R}(7)\ [\mathcal{R}8\mathcal{A}(8)\mathcal{A}(8)]\ \mathcal{A}(7)\ [\mathcal{R}8\mathcal{A}(8)\mathcal{A}(8)]\ \mathcal{A}(7)]\ \mathcal{A}(6)\ [\mathcal{R}(7)\ [\mathcal{R}8\mathcal{A}(8)\mathcal{A}(8)]\ \mathcal{A}(7)\ [\mathcal{R}8\mathcal{A}(8)\mathcal{A}(8)]\ \mathcal{A}(7)]\ \mathcal{A}(6)]$ Square brackets separate different levels of recursion A vivid description of the algorithm can be found in [6] In the procedure $\mathcal{A}(l)$ the fluxes $\widetilde{E}$ are calculated using one of the numerical schemes described in the previous chapter

3.3 Mesh Refinement

The most difficult part of adaptive mesh refinement is to decide where and when to refine or unrefine a mesh The refinement procedure currently implemented consists of four steps

1 For every cell, a refinement indicator, $0 \leq \xi \leq 1$, is computed Large $\xi > \xi_{split}$ indicates that a leaf must be refined, and small $\xi < \xi_{join}$ indicates that a split cell can be unrefined, ξ_{split} and ξ_{join} are some predefined constant values

2 Smooth ξ in order to prevent cells from being falsely refined (mesh trashing) in places where ξ fluctuates around critical values ξ_{split} and ξ_{join}

3 Leaves are refined if $\xi > \xi_{split}$

4 Split cells are unrefined if $\xi < \xi_{join}$ and if they have not just been split and if joining them does not produce an isolated leaf

An indicator proportional to the gradients in the solution can be used to compute ξ Such an indicator shows where to expect a large error in the solution The indicator ξ

is constructed as a maximum of several indicators,

$$\xi = \max\left(\xi^1, \xi^2, \quad\right) \tag{3 4}$$

each of which is either a shock indicator, a contact discontinuity indicator, or a gradient indicator, all normalized to unity, $0 \leq \xi^k \leq 1$ As a shock indicator the quantity used is,

$$\begin{aligned} \xi_i^s &= \max_{j=1,2} \xi_{i,j}^s, \\ \xi_{i,j}^s &= \begin{cases} 1 & \text{if } \dfrac{|p_{iNb(i,j)} - p_i|}{\min\left(p_{iNb(i,j)}, p_i\right)} > \varepsilon_s, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3 5}$$

The constant ε_s determines the minimum shock strength to be detected. In the present work, ε_s was generally set to 1 0 A contact discontinuity indicator is defined as,

$$\begin{aligned} \xi_i^c &= \max_{j=1,2} \xi_{i,j}^c, \\ \xi_{i,j}^c &= \begin{cases} 1 & \text{if } \dfrac{|p_{iNb(i,j)} - p_i|}{\min\left(p_{iNb(i,j)}, p_i\right)} < \varepsilon_s, \dfrac{|\rho_{iNb(i,j)} - \rho_i|}{\min\left(\rho_{iNb(i,j)}, \rho_i\right)} > \varepsilon_c \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3 6}$$

The quantity ε_c was set to 0 1 for most calculations Another possible indicator for the contact discontinuity is,

$$\xi_{i,j}^c = \begin{cases} 1 & \text{if } \dfrac{|\rho_{iNb(i,j)} - \rho_i|}{|p_{iNb(i,j)} - p_i|} > \varepsilon_c \\ 0 & \text{otherwise} \end{cases} \tag{3 7}$$

One more indicator ξ_i^z is used for locating the interface between the burnt gases and fresh propellants This indicator is constructed on the reactant mass fraction Z as,

$$\xi_i^z = \begin{cases} 1 & \text{if } 0\,02 < Z < 0\,98 \\ 0 & \text{otherwise} \end{cases} \tag{3 8}$$

A gradient indicator for a variable b may be constructed as

$$\xi_i^b = \max_{j=1,2} \left(\frac{||b_{iNb(i,j)}| - |b_i||}{\max\left(|b_{iNb(i,j)}|, |b_i|\right)} \right) \tag{3 9}$$

where b may be mass density, energy density, pressure, velocity, vorticity etc For the present problem the gradient indicator is constructed on pressure and temperature According to equations 3 4–3 8, shocks and contact discontinuities are marked with $\xi = 1$ and refined to the maximum refinement level allowed and the gradient indicators 3 9 are applied only to a smooth flow.

Chapter 4

Pulsed Detonation Engine Model

The Pulsed Detonation Engine (PDE) is a novel alternative to conventional gas turbine or rocket engines and may provide improvements in both performance and costs for space propulsion The aim of the present work is to develop a computational model of the PDE, which can simulate all stages of the working cycle The PDE model was developed based on a recent study by Mohanraj [10] The following sections present the operational details of the PDE model, along with various parameters used in controlling the operation The default values used for those parameters are also stated, and those were the typical values used for obtaining all the results of chapter 5, unless otherwise stated

4.1 PDE operational stages

The various stages in the operation of the PDE are shown in figure 4 1 A typical cycle of operation of the PDE can be divided into the following three stages

Detonation initiation and propagation At the beginning of every cycle, the tube is full of fresh fuel and oxidizer The detonation is initiated by specifying a finite

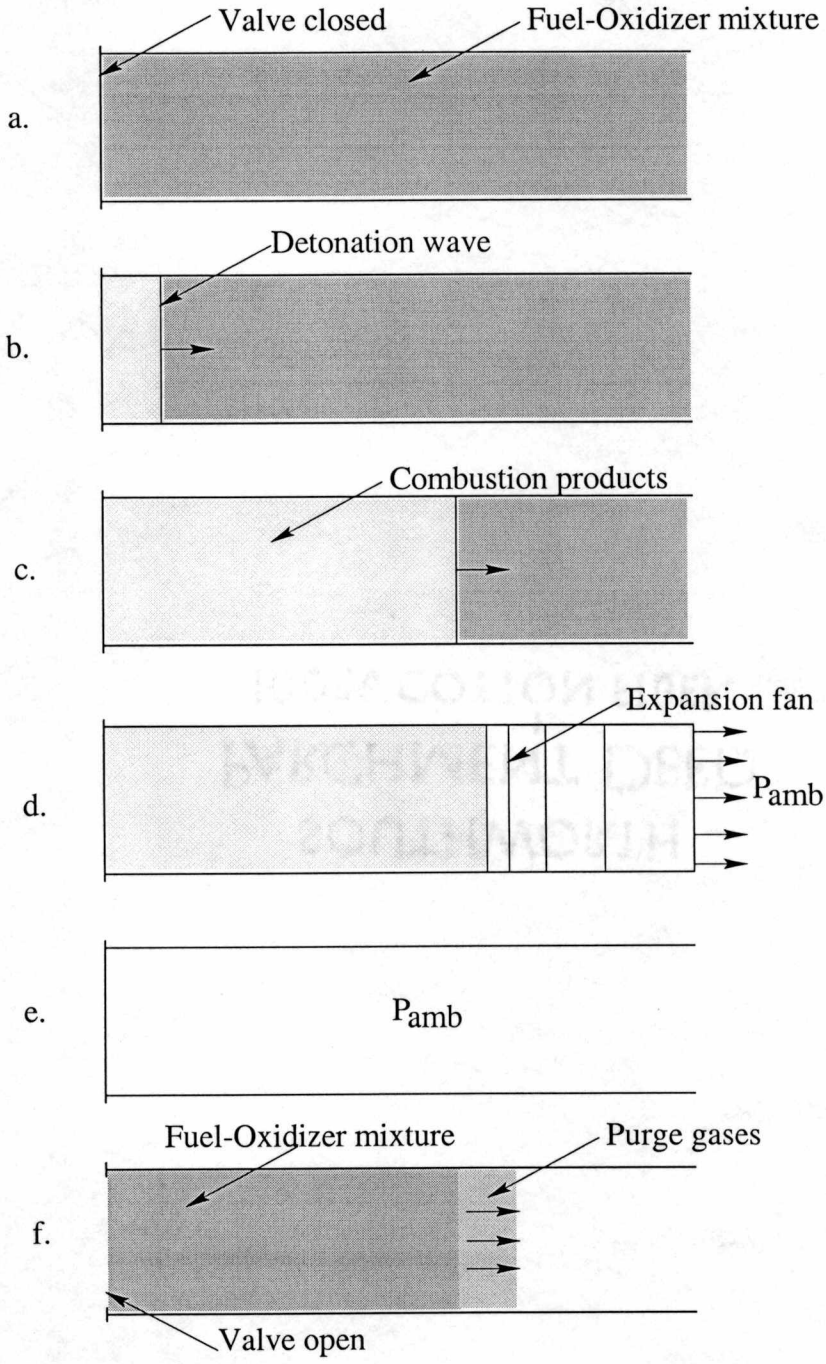


Figure 4.1: Various stages in the operation of a pulsed detonation engine. (a) Tube filled with fuel and oxidizer; (b) Detonation initiation; (c) Detonation propagation; (d) Blow-down phase; (e) Tube after the combustion products have been exhausted; (f) Propellant fill phase.

region of high temperature and pressure at the upstream end of the tube A typical initiation region would be of size $x_{init} = 0.002m$, with temperature $T_r = 3500K$ and pressure ratio $p_r = 30$ Once initiated the detonation wave traverses through the tube consuming the reactants

Blowdown phase After the exit of the detonation wave the bulk of the combustion products are expelled by a system of compression and rarefaction waves

Propellant fill phase In the present model, the time period of cyclic operation of the PDE is specified, and the valve opening is determined by this time period The injection of fresh propellants immediately behind the burnt gases can result in premature heating and combustion of the new propellant charge during the filling phase This is prevented by injecting sufficient amount of purge gases before the propellant charges The injection of propellants is stopped by closing the valve at the upstream end, as soon as the fresh charge of propellants reaches a sensor location (close to the end of tube) The end of the filling phase is followed by initiation of a fresh wave of detonation and the cyclic operation continues

4.2 Computational Model

The one-dimensional reactive Euler equations 2 1 are used to represent the flow processes in the PDE model The reaction rate is modelled by a one step irreversible reaction equation governed by Arrhenius kinetics The working fluid is a stoichiometric $H_2 - O_2$ mixture, the values of various parameters used for this working fluid are as follows

Ratio of specific heats $\gamma = 1\ 25$, molecular weight $MW = 15kg/kmol$, activation energy $E_a/R_u = 1500K$, pre-exponential factor $K = 7.5 \times 10^9$, heat release per unit mass $q_0 = 7\ 769 \times 10^6 J/kg$

The PDE engine geometry used for the results presented in the next chapter are based on a constant area duct, open at one end, and with a valve at the other However the present quasi one-dimensional framework leaves scope for specifying any arbitrary area distribution along the tube length and for the study of various geometries

4.3 Boundary conditions

The upstream boundary is modelled as a valve, which remains open while the tube is being filled, and is closed at all other times When the valve is fully closed, the upstream boundary is modelled as a rigid wall, by applying mirror conditions over a ghost cell lying across the wall When the valve is open or partially open, the inlet boundary conditions are obtained from the specified reservoir conditions For low values of the reservoir pressure, the stagnation pressure at the inlet boundary is the same as the reservoir pressure If the reservoir pressure is above the choking limit, the stagnation pressure (in conjunction with the instantaneous valve area) is used to specify the mass flow rate To simulate realistic valve operation in the PDE model, the valve opening and closing processes are characterized by a valve response time (τ_{valve}). During the valve opening process the valve area is linearly increased from zero (fully closed) to the specified maximum (fully open) valve area, and vice versa during the valve closing process In the present work, instantaneous valve action was assumed ($\tau_{valve} = 0$) and

the reservoir pressure was set to $5 \times 10^5 N/m^2$ The maximum valve area was chosen as ten percent of the cross-sectional area of the tube

The downstream boundary can have several possible outflow conditions subsonic inflow or outflow, or sonic or supersonic outflow Sonic and supersonic inflow could be allowed, but has not been implemented Under some operating conditions, the back pressure is so low that choked outflow conditions prevail over most of the cycle For high back pressure conditions, the exit unchokes and subsonic outflow or reverse flow may occur The exit boundary conditions are based upon the method of characteristics procedure If the outflow is sonic or supersonic, no exit boundary condition is allowed, and all variables at the exit plane are determined from conditions inside the flowfield Sonic or supersonic outflow generally implies the pressure at the exit plane is above the ambient pressure If the outflow is subsonic, the exit pressure is specified as the ambient pressure and the remaining quantities are computed using the characteristic equations corresponding to the outgoing waves If reverse flow occurs, the stagnation values for pressure and temperature are specified as those corresponding to the ambient condition, while the reactant mass fraction Z is taken as 0 Typical values for the ambient conditions are $p_{amb} = 0\ 5 \times 10^5$ and $T_{amb} = 300K$

The procedure for switching from sonic outflow to subsonic outflow is important because the sonic outflow condition does not involve external conditions Consequently, once the outflow becomes sonic, there is no mechanism for checking when it should unchoke again The present model employs a procedure to determine whether the pressure at the nozzle exit is such that the presence of a normal shock results in a post-shock

pressure lower than the ambient pressure If that were the case, the boundary condition forces a normal shock at the exit and thereafter enforces the subsonic outflow condition

Chapter 5

Results and Discussion

5.1 Detonation Initiation and Mesh Size

The aim of the present study is to perform an adaptive mesh refined simulation of the detonation phenomenon to capture the finer aspects of the flow with higher precision However, the dependency of the initiation process on the mesh size, needs to be studied to analyze the effect of mesh refinement Numerical simulations were performed using Roe's flux equation 2 9 for a tube of length $L = 0\,05m$ with an initiation region of size $x_{init} = 0\,002m$ In the initiation region the pressure ratio p_r used was 30 and the temperature T_r was set to 3500K Simulations were carried out on uniform meshes of size 256, 512, 1024 and 2048 cells The initiation is a physical process, which is being modelled numerically in these computations The cell size Δx is a part of the computational model and it influences the initiation process

While the detonation could be successfully initiated in the first 3 cases, it failed to initiate for meshes of size 2048 cells and above A comparison of figures 5 1 and 5 2 shows the difference between a case in which the detonation initiates properly and one in

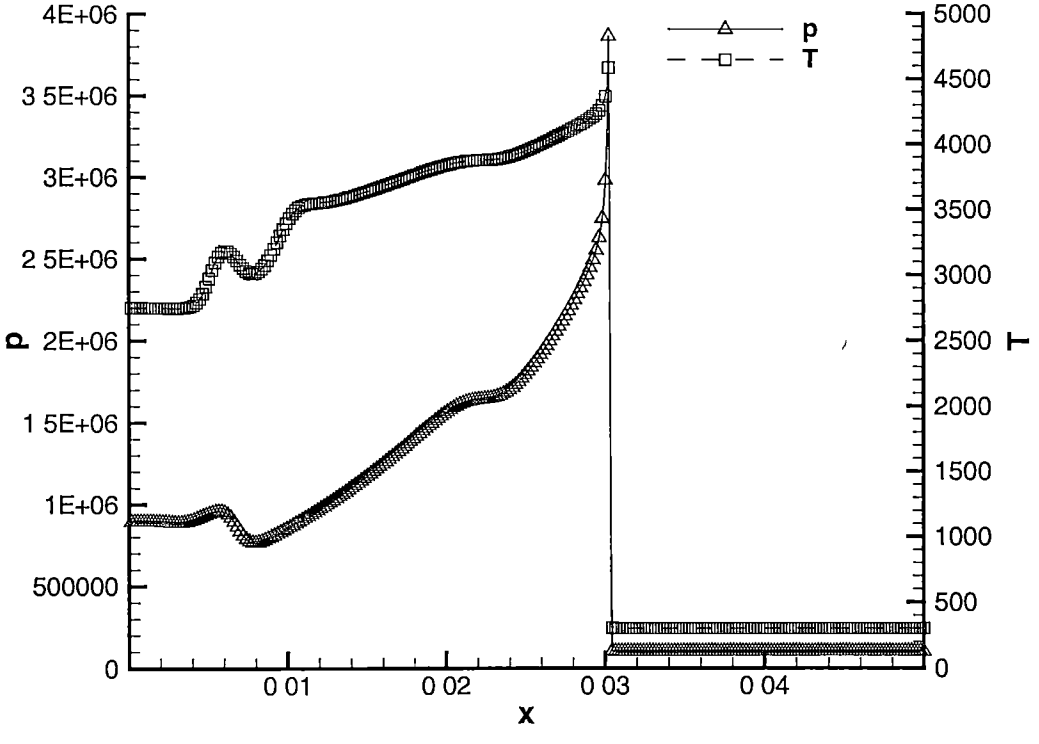


Figure 5 1 Pressure and Temperature profiles at $t = 1\ 0 \times 10^{-5} s$, for a uniform mesh of 1024 cells (Every 4th point is shown)

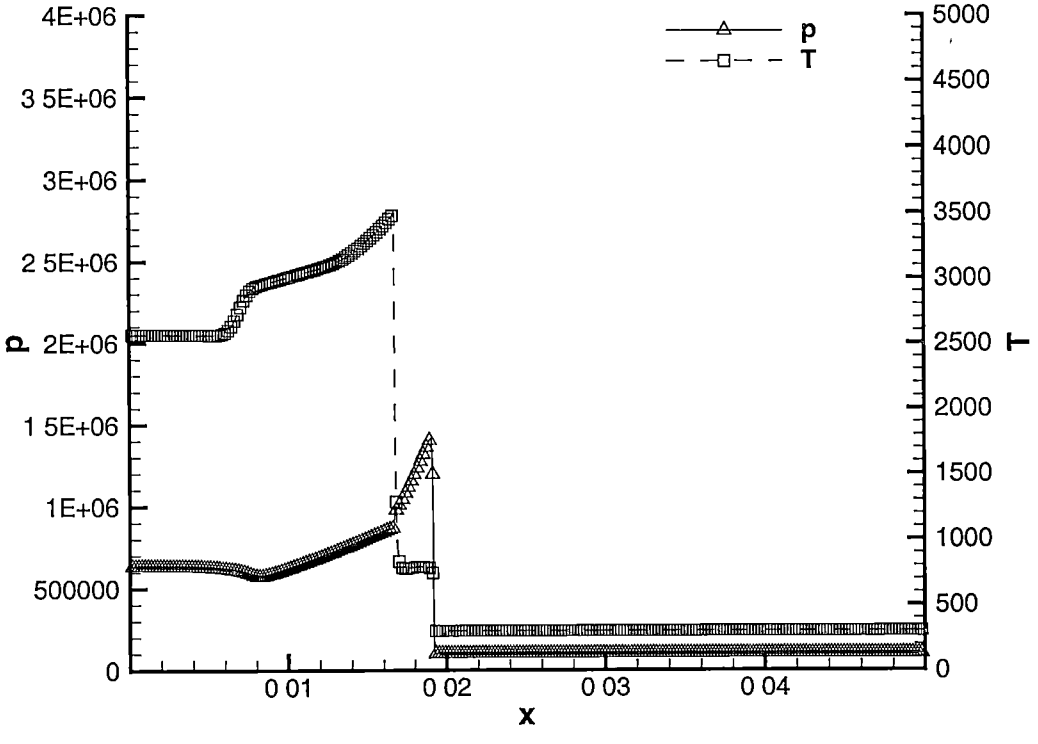


Figure 5 2 Pressure and Temperature profiles at $t = 1\ 0 \times 10^{-5} s$, for a uniform mesh of 2048 cells (Every 8th point is shown)

which it fails to initiate In figure 5 1 the shock and the flame are co-located Figure 5 2 shows the shock running ahead of the flame Also note that the shock in figure 5 2 has travelled a much shorter distance than the one in figure 5 1 (the figures correspond to the same time) The reduced numerical diffusion associated with smaller values of Δx might be one reason why the detonation fails to initiate in fine meshes If the temperature of a precursor cell is high, numerical diffusion would raise the temperature in the next cell to the ignition limit or above and help in the propagation of the reaction front It might be this factor that is missing, when Δx is reduced below a limit To discount the possibility of a spurious propagation of the detonation wave due to the suspected role of numerical diffusion, the observed speed of detonation wave is compared with the theoretically predicted speed

The detonation velocity v_D, may be defined as the velocity at which the detonation wave enters the unburnt fuel mixture As shown in [15] an approximate expression for the detonation velocity would be,

$$v_D = \left[2\left(\gamma_2 + 1\right)\gamma_2 R_2 \left(\frac{C_{p,1}}{C_{p,2}} T_1 + \frac{q_0}{C_{p,2}}\right)\right]^{1/2} \tag{5 1}$$

where T_1 is the temperature of the unburnt fuel and $C_{p,1}$ and $C_{p,2}$ are the specific heats of the reactants and products For the present case, $C_{p,1} = C_{p,2}$ and the equation simplifies to,

$$v_D = \left[2\left(\gamma + 1\right)\gamma R \left(T_1 + \frac{q_0}{C_p}\right)\right]^{1/2} \tag{5 2}$$

Substituting $\gamma = 1\ 25$, MW=15, $R_u = 8314 J/(kg - K)$, $R = R_u/MW$, $q_0 = 7\ 769 \times$

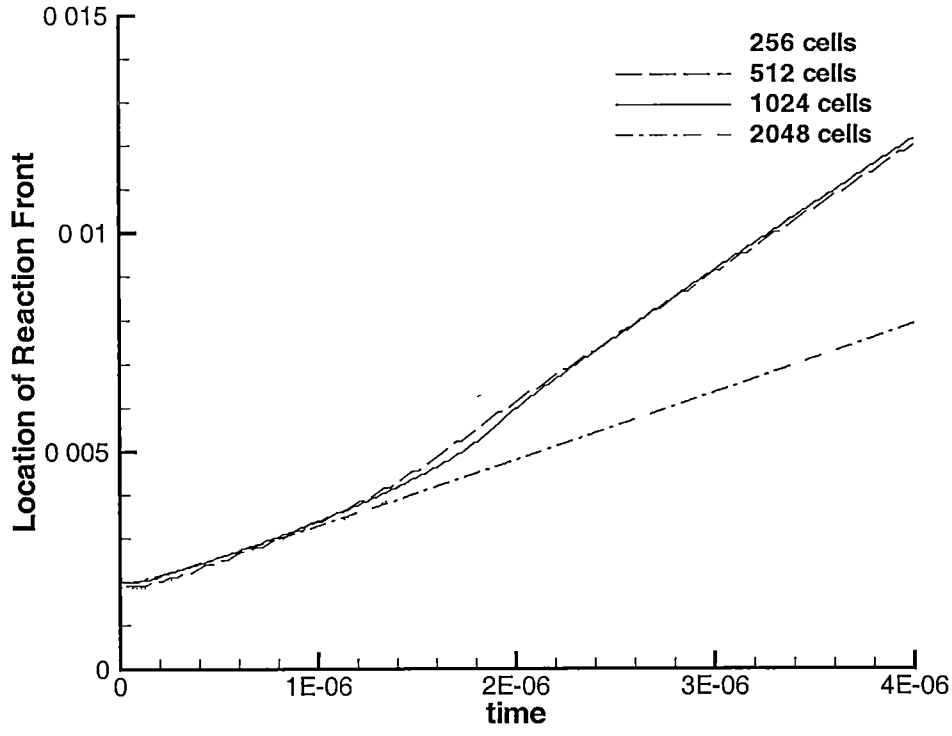


Figure 5 3 Location of Reaction front as a function of time for various mesh sizes during the initiation phase

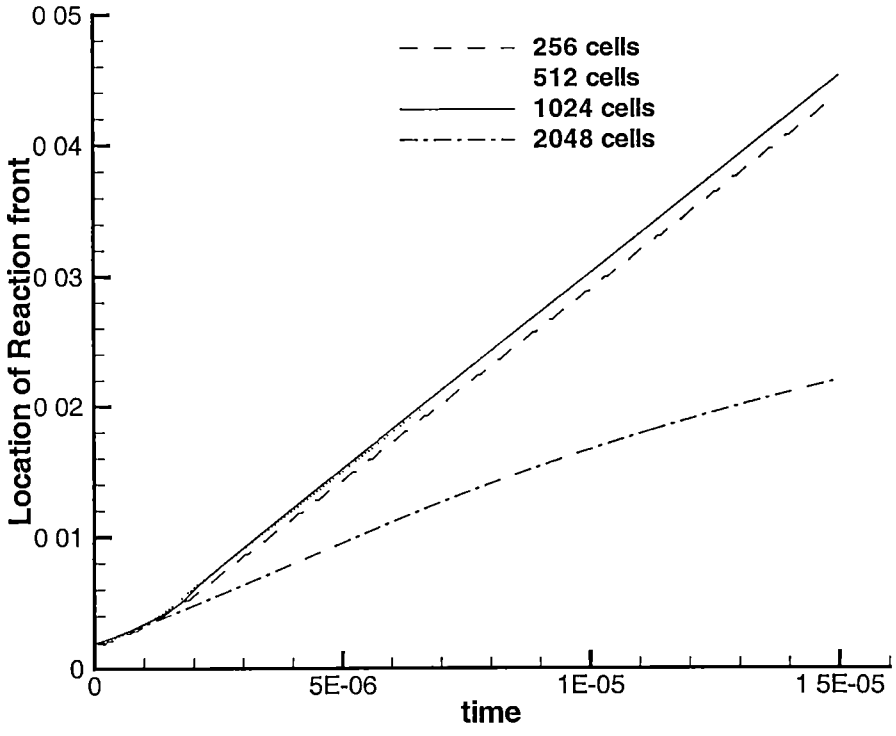


Figure 5 4 Location of Reaction front as a function of time for various mesh sizes on a longer time scale

$10^6 J/kg$ and $T = 300K$, the detonation velocity v_D is found to be 3110 54m/s Figures 5 3 and 5 4 show the distance traveled by the reaction front with respect to time for various mesh sizes The observed speed of the reaction front is $3015m/s$ and is almost the same for all the cases in which the detonation was initiated properly

Figures 5 5—5 8 show the other notable difference between the solutions obtained for various mesh sizes The peak pressure and temperature (just behind the reaction front) are higher for smaller cell sizes and there is a big difference in the peak pressure and temperature values reached during the initiation phase These could be explained only with considerable insight about the initiation and propagation mechanism

An indication of the complexity of the flow dynamics during the initiation period can be seen from the $x - t$ diagrams 5 9 and 5 10 They correspond to the results shown in figures 5 1 and 5 2 The $x - t$ diagrams show details of the shock, expansion fan and contact surface The pathline of the shock-wave in figure 5 9 shows an increase in velocity when the detonation was initiated This is in contrast to figure 5 10, where the detonation fails to initiate and the shock continues to move at the same velocity

5.2 Contact Discontinuity

During a typical application of the pulsed detonation engine model, the initiation and propagation of the detonation wave are simulated followed by the ensuing blow-down process This is followed by the propellant fill process It is during this stage that the accurate capture of the contact discontinuity plays a significant role As explained in the previous chapter, during the filling process a buffer region of non-combustible purge gases

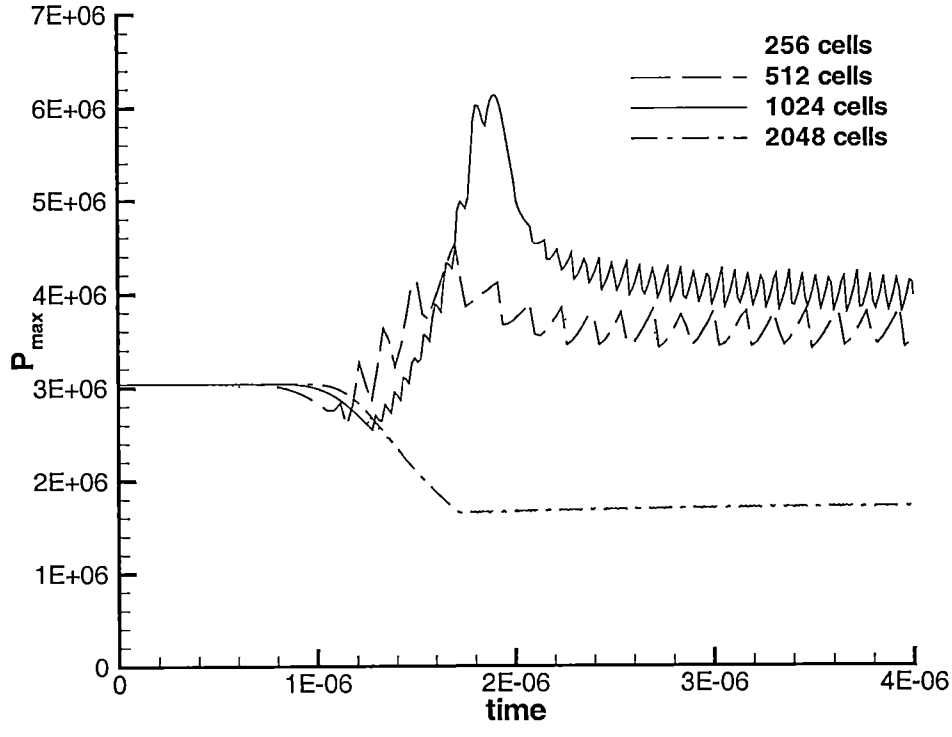


Figure 5 5 Comparison of peak pressure for various mesh sizes during the initiation phase

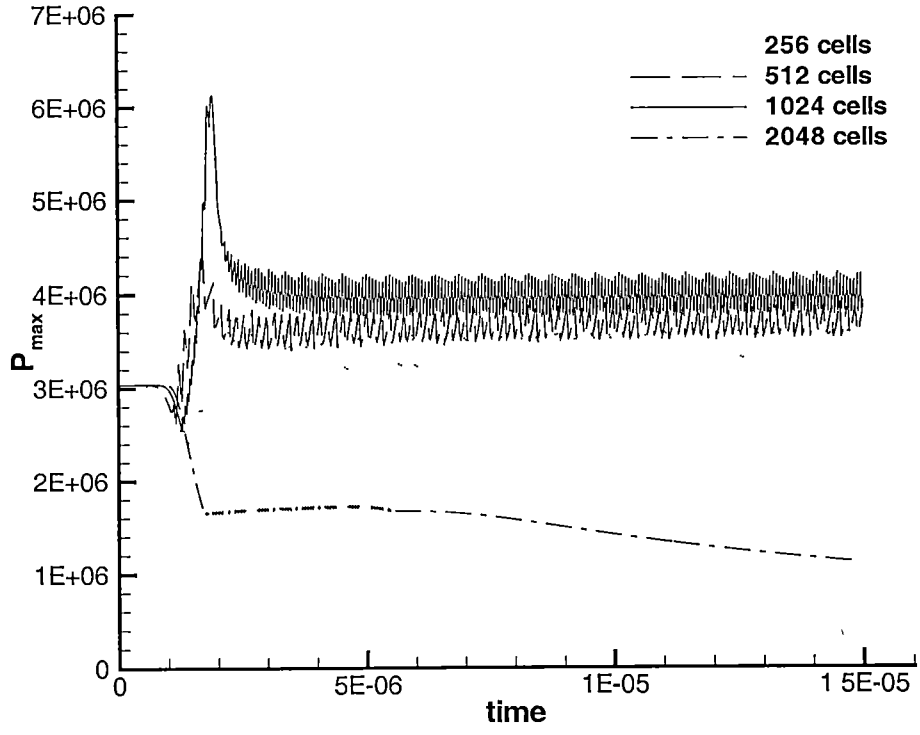


Figure 5 6 Comparison of peak pressure for various mesh sizes on a longer time scale

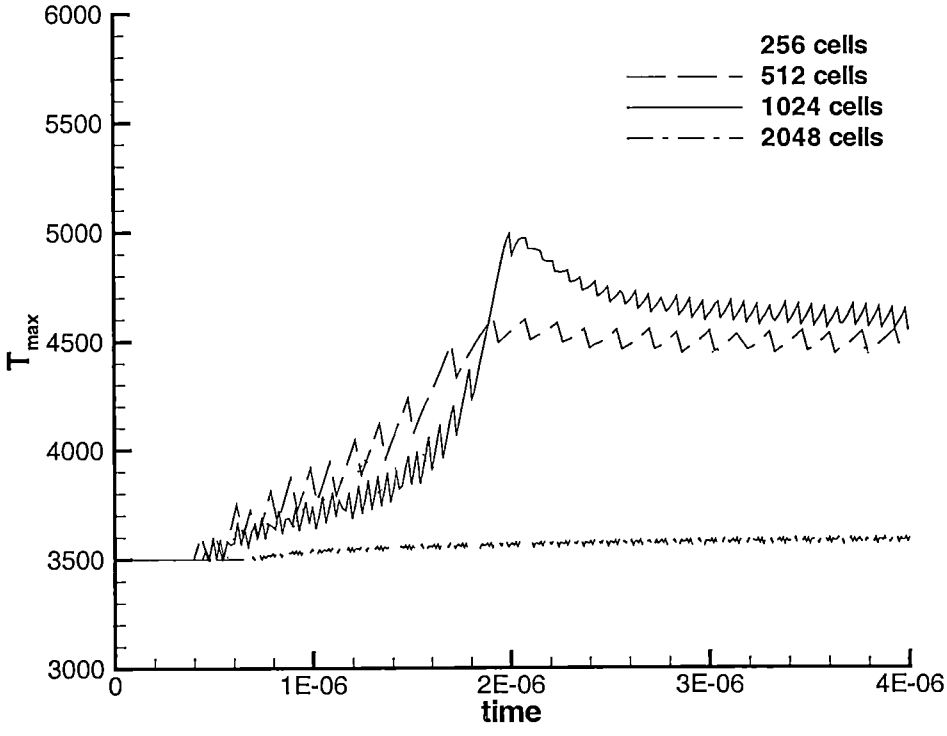


Figure 5 7 Comparison of peak temp for various mesh sizes during the initiation phase

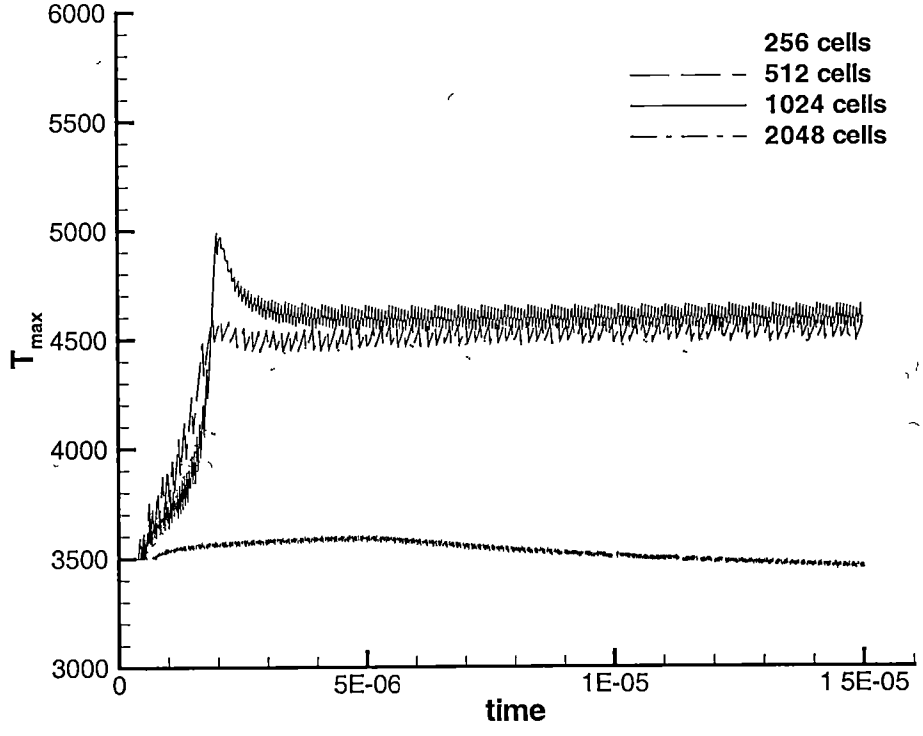


Figure 5 8 Comparison of peak temp for various mesh sizes on a longer time scale

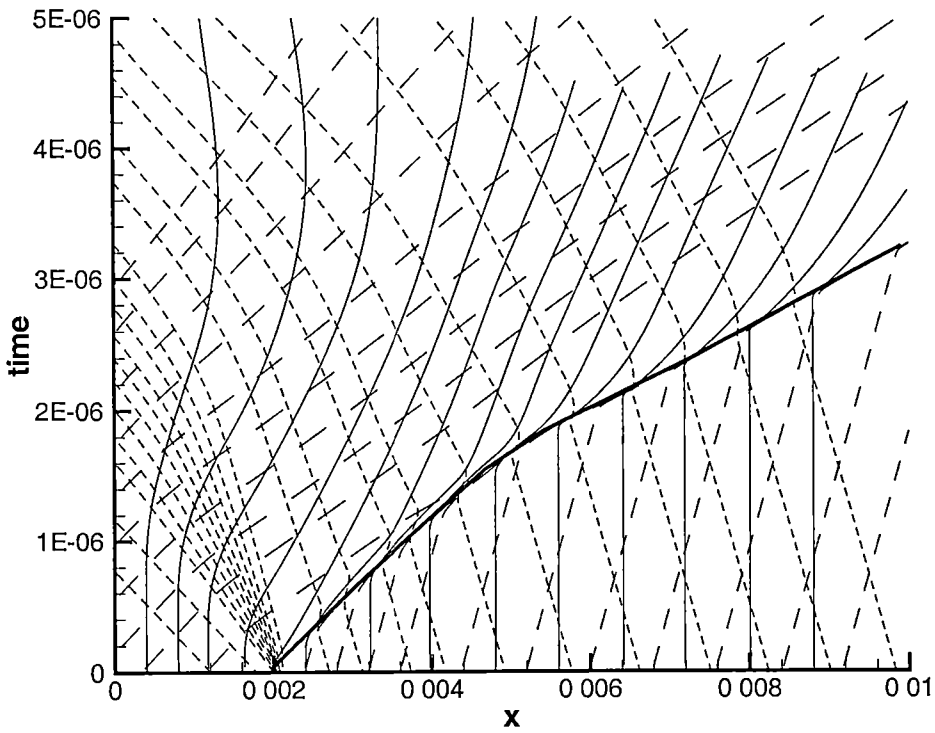


Figure 5 9 $x - t$ diagram during the initiation period for a uniform mesh of 1024 cells (a) ——— Pathline, (b) ——— Pathline of shock-wave, (c) — — — Right running (u+a) Mach wave, (d) - - - - - - Left running (u-a) Mach wave

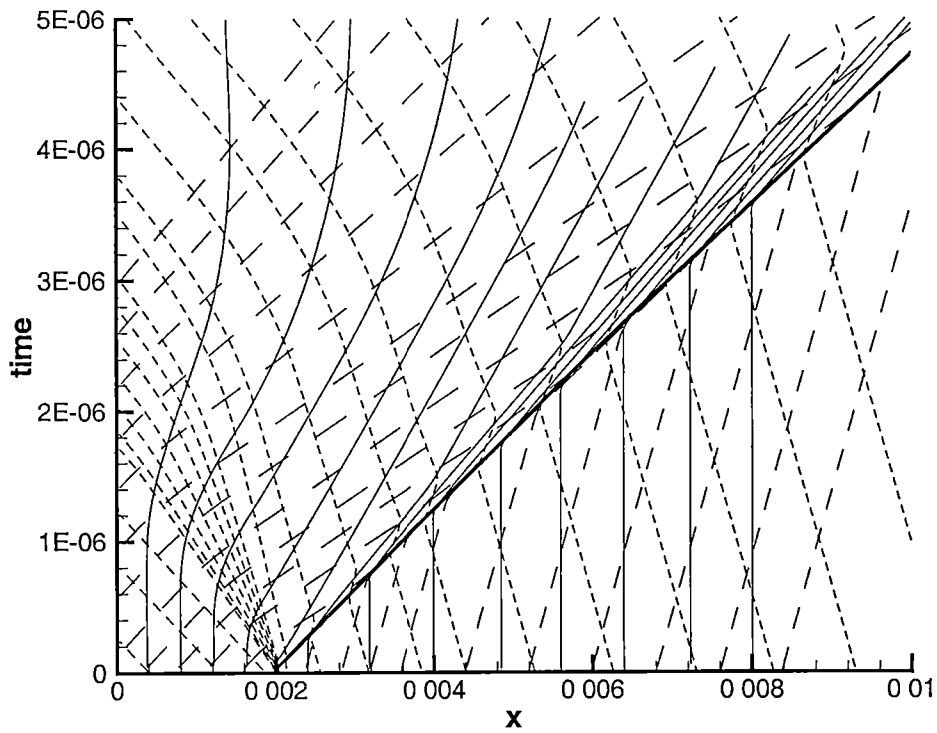


Figure 5 10 $x-t$ diagram during the initiation period for a uniform mesh of 2048 cells (a) ———— Pathline, (b) ———— Pathline of shock-wave, (c) — — — Right running (u+a) Mach wave, (d) - - - - - - Left running (u-a) Mach wave

is injected first followed by fresh propellants The presence of purge gases avoids the premature heating and combustion that could have occurred had the fresh propellants been injected immediately behind the burnt gases During this phase, there are two contact surfaces that need to be captured accurately with minimal numerical diffusion, one between the cold purge gases and the burnt gases and the other between the fresh propellants and the purge gases

Roe's approximate Riemann solver (equation 2 9) has been known to smear contact discontinuities, as do most other shock-capturing schemes Figure 5 11 shows the interface profiles, at one instant of time during the filling phase, computed using three different mesh sizes (256, 512 and 1024 cells) The plot shows the profiles of both the density (ρ) and the reactant mass fraction (Z) Decreasing the mesh size, does reduce the thickness of the interface, but it becomes smeared over a larger number of points Whenever Δx is reduced, the permissible time step per iteration is also reduced Hence, the solver does a larger number of time steps (approximately 4 times as many) on a mesh with 1024 cells, than on a mesh of 256 cells This gives more time for the interface to diffuse across more cells for a given physical time But the smaller mesh size does help to capture the interface with slightly better accuracy

From results obtained for a mesh of 256 cells, (see figure 5 11) it can be seen that the unburnt fuel has diffused far enough to come in contact with the burnt gases even before it has travelled half the tube length This high numerical diffusion offsets the purpose of injecting purge gases Refining the mesh could solve this problem to some extent, but the effect is quite slow as noted here Figure 5 12 shows a plot of the interface thickness

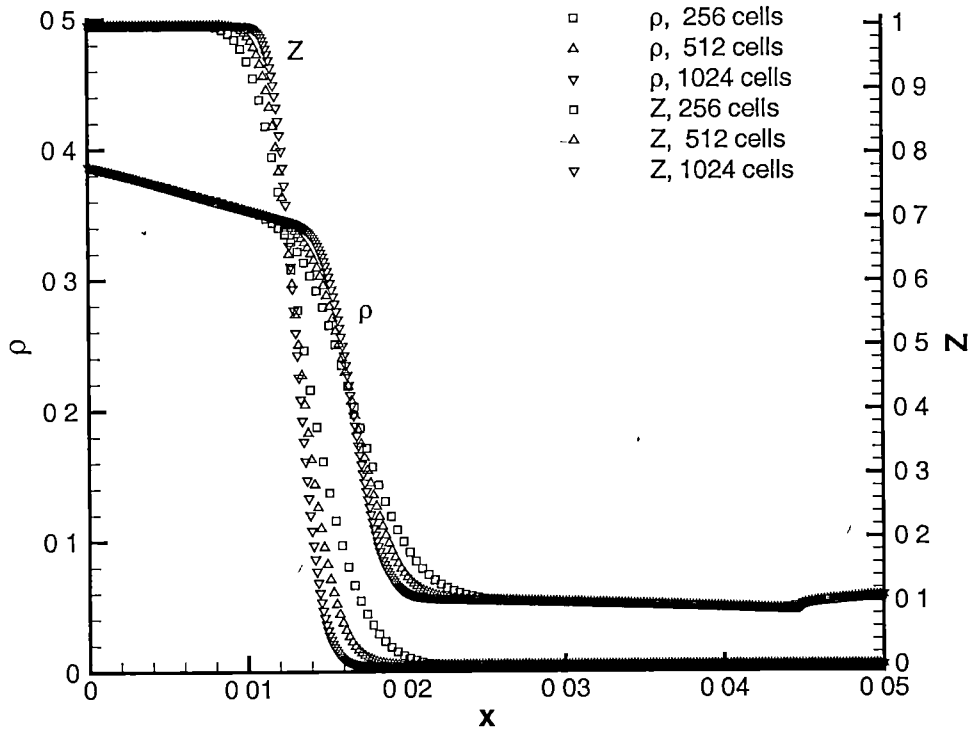


Figure 5 11 Density (ρ) and Reactant Mass fraction (Z) profiles at time $t = 1\,8 \times 10^{-4}$s, for different mesh sizes (Every other point is shown)

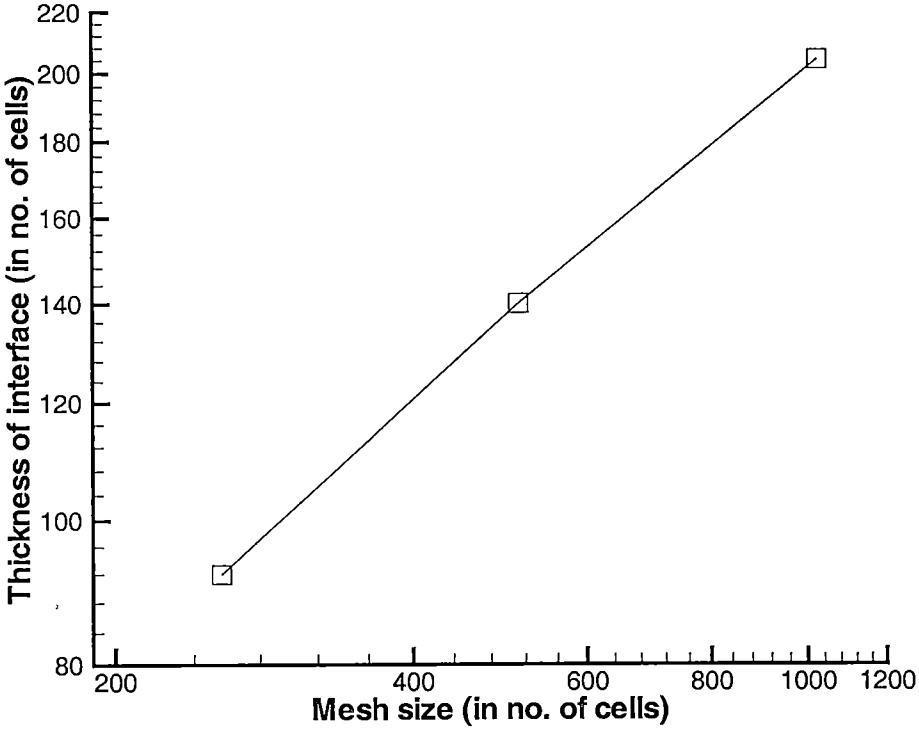


Figure 5 12 Plot of thickness of interface vs mesh size

with respect to the mesh size in number of cells The interface spreads to a greater number of cells as the mesh is refined further This is an indicator to the fact that the numerical scheme is less than first order accurate in capturing a contact surface

Figure 5 13 presents a comparison of contact surface profiles, obtained using three different numerical schemes, Roe's scheme, AUSM$^+$(both are explained in chapter 2) and the space-time method [2] The space-time method is second order accurate in space and time, while the other two are first order accurate The space-time method is able to capture the contact surface with reasonable accuracy and the solution is far better than those obtained with the other two schemes Comparing the first order Roe and AUSM$^+$ schemes with a second order solution is not a good way to estimate the capability of these schemes, but this comparison helps to verify the viability of space-time method One should note the close agreement between the results obtained using Roe's scheme and AUSM$^+$ The AUSM$^+$ scheme performs as well as Roe's scheme and in typical computations consumed 50% less CPU time The CPU time used by the different numerical schemes for typical simulations of a full PDE cycle on an Origin 2000 processor are listed in table 5 1 The table shows the significant cost advantage of using AMR (results from AMR are presented in the next section) Despite being faster than Roe's scheme, the AUSM$^+$ scheme is considerably slower than the space-time method Taking advantage of the efficiency of AUSM$^+$ over Roe's scheme, results for the later sections will computed using the AUSM$^+$ scheme

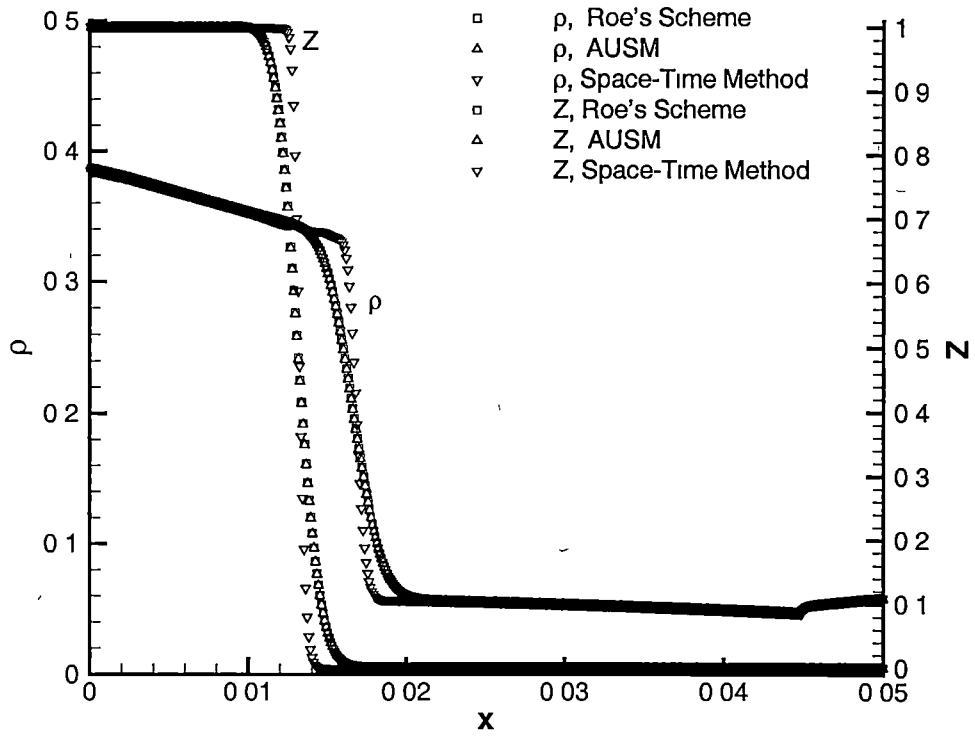


Figure 5 13 Comparison of Density (ρ) and Reactant Mass Fraction (Z) profiles at time $t = 1\ 8 \times 10^{-4}$s, obtained using different numerical schemes, for a mesh of 1024 cells (Every other point is shown)

Table 5.1· CPU time (in seconds) used by different numerical schemes for different mesh sizes on an Origin 2000 processor

Mesh Size	Space-Time Method	Roe's scheme	AUSM$^+$ scheme
256 cells	65	172	95
512 cells	263	692	378
1024 cells	1066	2774	1520
AMR, $l_{min} = 7$, $l_{max} = 10$	—	190	106

5.3 Adaptive Mesh Refined Simulations

The previous section shows the importance and effects of increasing the spatial resolution in the numerical simulation A cost effective way of increasing the spatial resolution is to finely refine the mesh wherever more resolution is required and let the coarse mesh remain where the cell size is inconsequential In this section we present results obtained with adaptive mesh refinement, using the Fully Threaded Tree (FTT) algorithm explained in chapter 3 As explained earlier, one of the important features of this algorithm is its ability to dynamically readjust the mesh as the flow evolves This makes sure that the fine aspects of the flow do not escape the finely refined region of the mesh

Figure 5 14 shows the evolution of the mesh as the solution is advanced in time The plot shows the pressure distribution at various instants of time, while the detonation wave is propagating along the length of the tube Also plotted is the mesh structure at those instants Recollecting the relationship between cell size Δ_i and level $iLv(i)$, $\Delta_i = 2^{-iLv(i)}L$, the cells that are highly refined are those whose level is equal to l_{max} The figure shows a case where $l_{min} = 7$ and $l_{max} = 10$ (Note $l_{max} = 10$ corresponds to 1024 cells when uniform grid is used) The following numerical values were used for the parameters that govern the refinement process, $\xi_{join} = 0\ 1$, $\xi_{split} = 0\ 4$, $\varepsilon_s = 1\ 0$ and $\varepsilon = 0\ 1$

The ambient pressure p_{amb} was set to $0\ 5 \times 10^5 N/m^2$ and the reference pressure p_{ref} to $1\ 01325 \times 10^5 N/m^2$ which resulted in a high pressure gradient at the exit during the onset of the computation This is one instance where the gradient indicator (equation 3 9) senses the high pressure gradient and the mesh is refined (as seen from the mesh structure

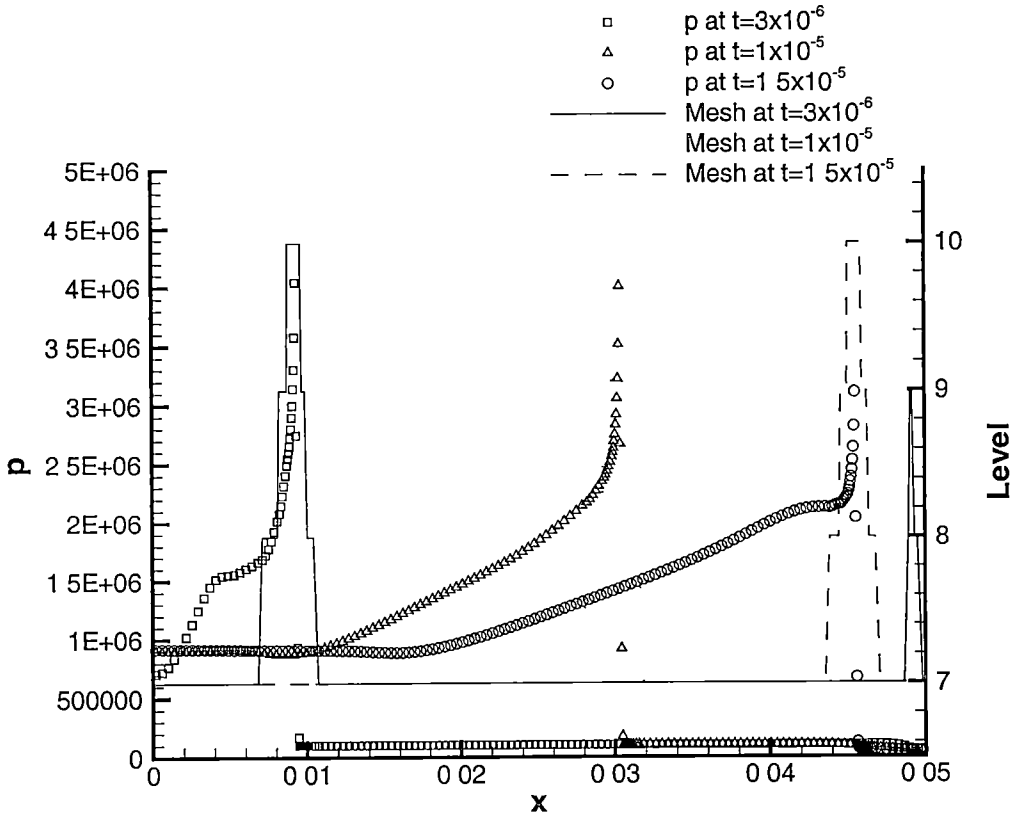


Figure 5 14 Pressure plot and mesh structure at three different instants of time during the propagation of the detonation wave $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,1$

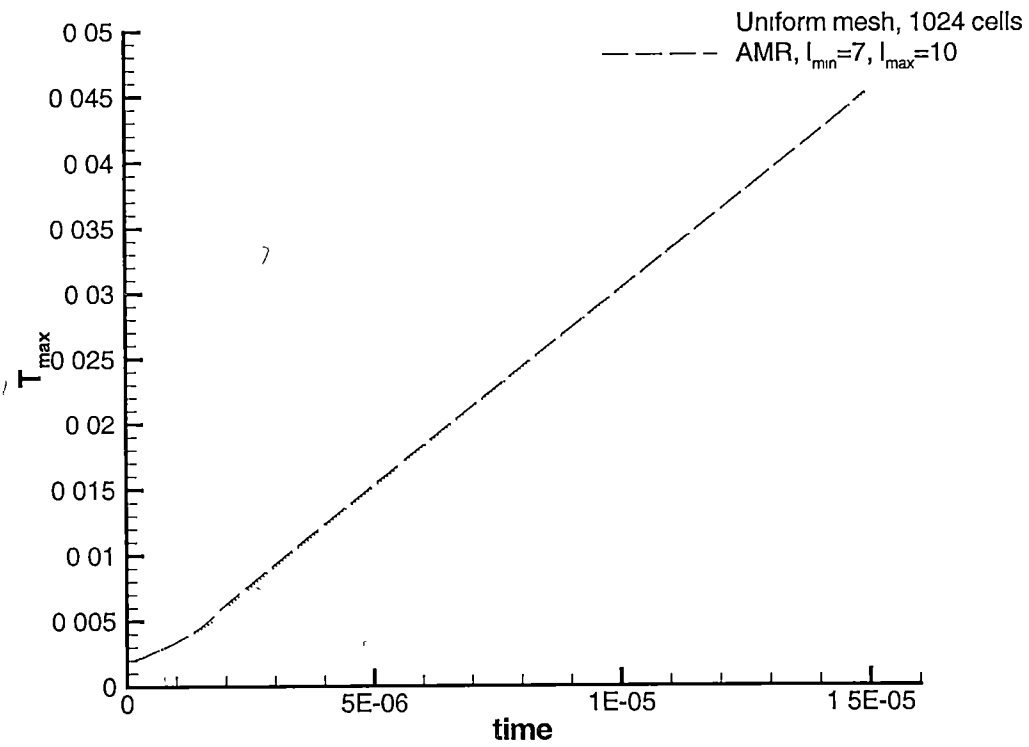


Figure 5 15 Location of reaction front as a function of time for a uniform grid and AMR grid

at $t = 3\times10^{-6}$s in figure 5 14) The location of the detonation front is sensed by the shock indicator (equation 3 5) and the finely refined region follows it continuously throughout the length of the tube Figure 5 15 shows the location of the reaction front with respect to time for an equal spaced mesh (1024 cells) and AMR mesh ($l_{min} = 7$ and $l_{max} = 10$) The location and speed of the detonation wave front obtained using AMR simulation agree closely with those obtained using a uniform grid

The discontinuity indicator (equations 3 6, 3 8) becomes significant during the filling phase, when the location of the interface needs to be sensed and the mesh refined at those locations However the contact surface was not sensed during the detonation propagation phase as evident in figure 5 14 Figures 5 16–5 18 show the density and reactant mass fraction plots, along with the mesh structure as the filling proceeds When the filling begins, the interface between the purge gases and the burnt gases is quite sharp and the indicator (equation 3 6) is able to locate its presence However as the interface moves and spreads to a few more cells, the indicator does not identify it as a discontinuity anymore and the mesh becomes coarse

The sensing of the Z-front is totally different The Z-sensor does identify the concentration gradient and then refines the mesh throughout the filling phase It can be inferred from equation 3 8 that the indicator will sense the Z-front and refine the mesh, even if it starts spreading to a large number of cells But a diffuse Z-front necessitates mesh refinement over a wide range of cells, which becomes computationally expensive

The interface sensing constant ε^c, plays a significant role in sensing an interface, reducing its value could be helpful in keeping track of an interface even when it starts

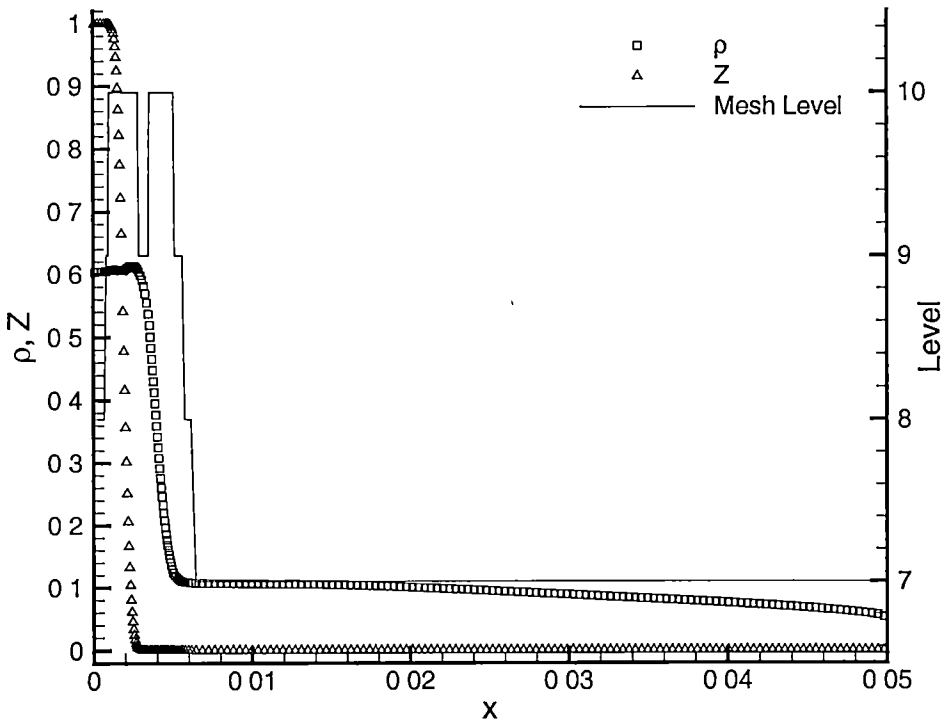


Figure 5 16 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,3 \times 10^{-4} s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,1$

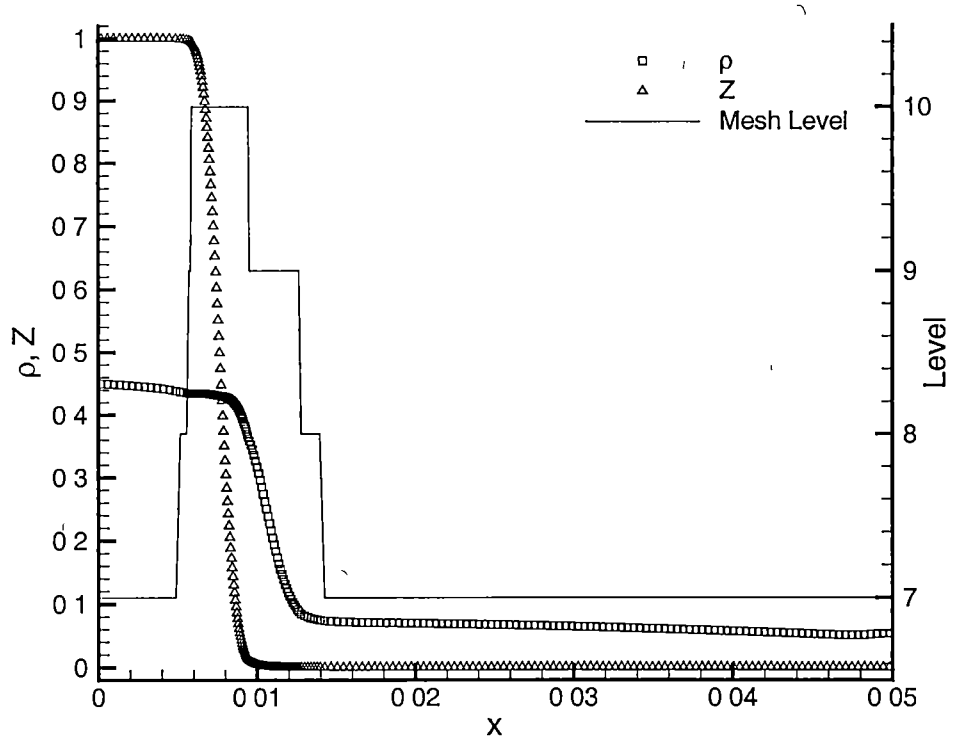


Figure 5 17 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,6 \times 10^{-4}s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,1$

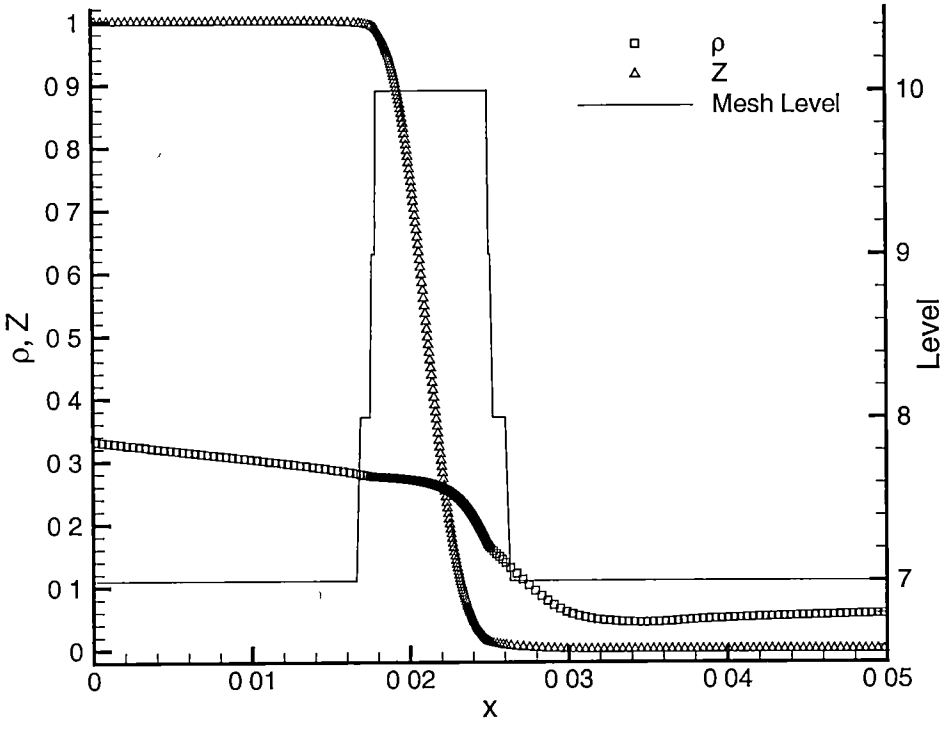


Figure 5 18 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\,0 \times 10^{-4}s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,1$

to diffuse However using a very low ε^c might lead to spurious mesh refinement Results obtained by reducing the value of ε^c from 0 1 to 0 05 are shown in figures 5 19–5 21 The results show that using a low ε^c helps to maintain a finely refined mesh around the interface for a longer time, but as noted earlier, this is of little help Figure 5 22 shows a comparison of the interface profiles obtained using a uniform mesh (1024 cells) and AMR mesh ($l_{min} = 7$ and $l_{max} = 10$) The Z-front on the AMR mesh is as wide as it was on the uniform mesh However the density plot shows that the contact-surface between the hot and cold gases in the AMR grid, has diffused to a greater number of points than in the uniform grid This is because the discontinuity indicator fails to identify the interface once it starts diffusing and the mesh becomes coarse near the tail end of the interface This is in contrast to the Z-interface which was sensed properly (even when diffuse) and captured as accurate as a uniform grid simulation

Other than using a contact discontinuity indicator, there is one more way to track an interface and refine the mesh For simple problems like the present one, it is possible to introduce a tracking variable x_{if}, whose initial value is set during the beginning of the filling phase As the solution is advanced during each time step, the location of the interface is updated using the interface velocity u_{if} (velocity at the point closest to x_{if}) at that instant

$$x_{if}^{new} = x_{if}^{old} + u_{if}\Delta t\,(l)$$

It is now possible to create a band, with its center at x_{if} and thickness Δx_{if}, within which the discontinuity indicator ξ_i^c is set to unity This allows the cells within that band to be refined to the maximum level Figures 5 23–5 25 show the results obtained

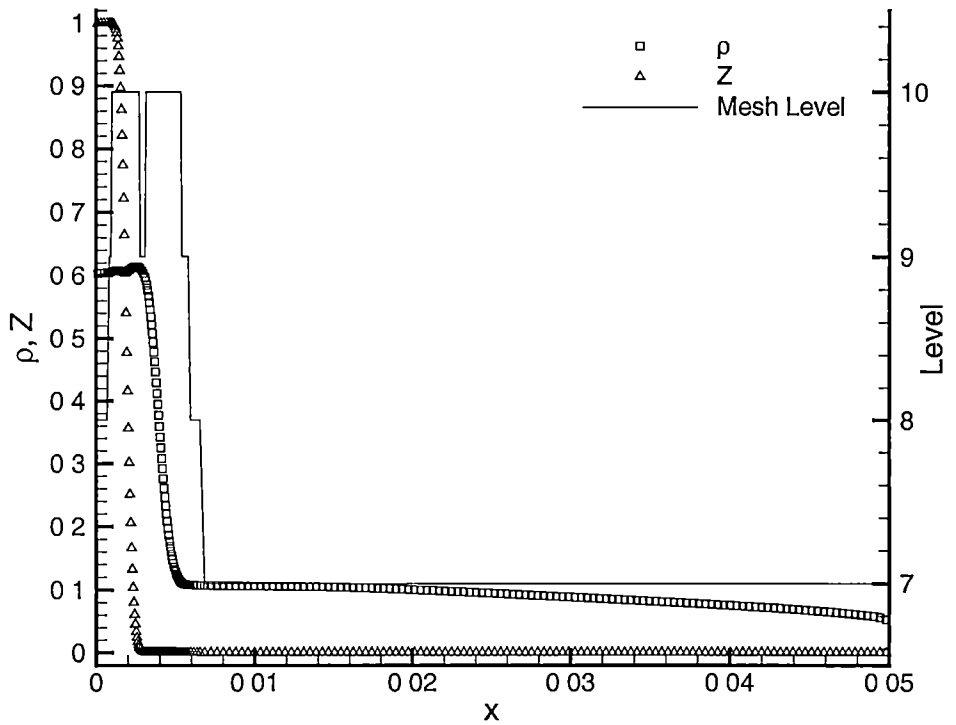


Figure 5 19 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,3 \times 10^{-4} s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$

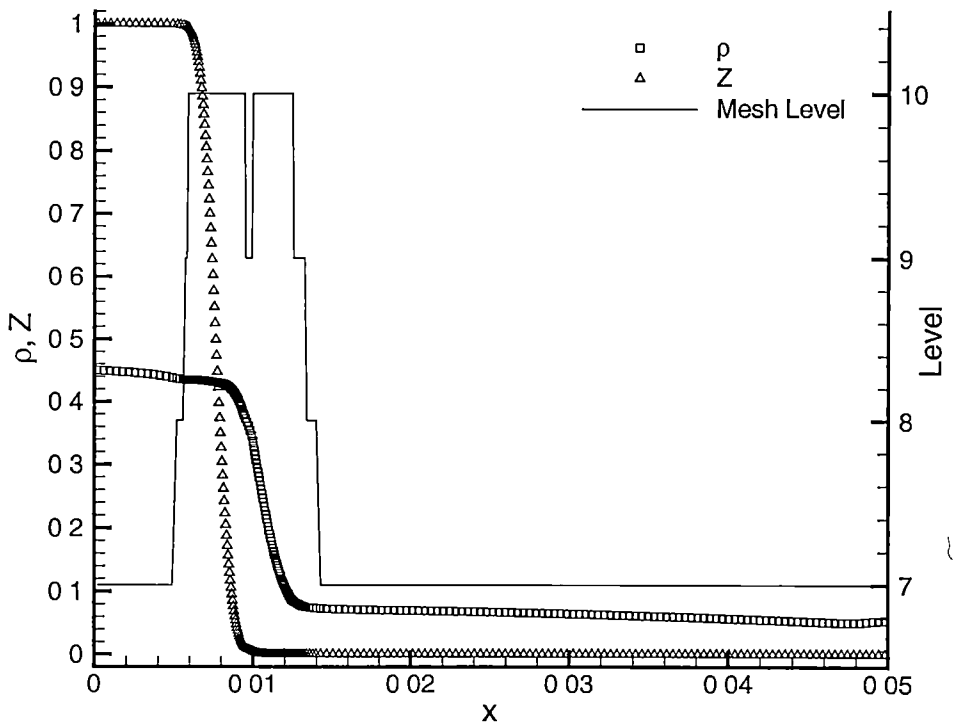


Figure 5 20 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,6 \times 10^{-4} s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$

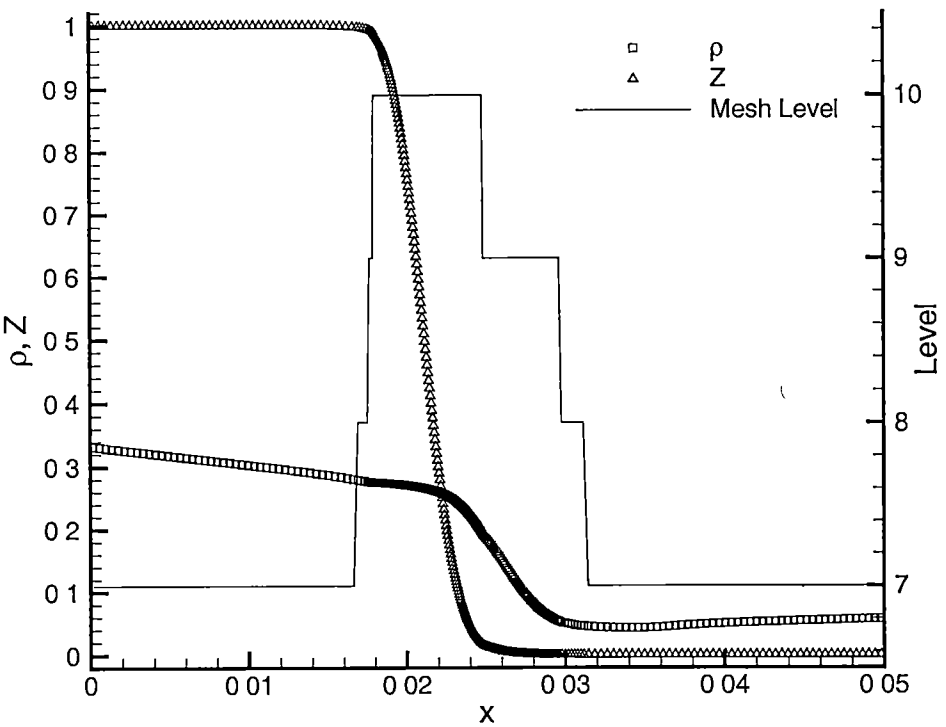


Figure 5 21 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\,0 \times 10^{-4} s$, during the filling phase $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$

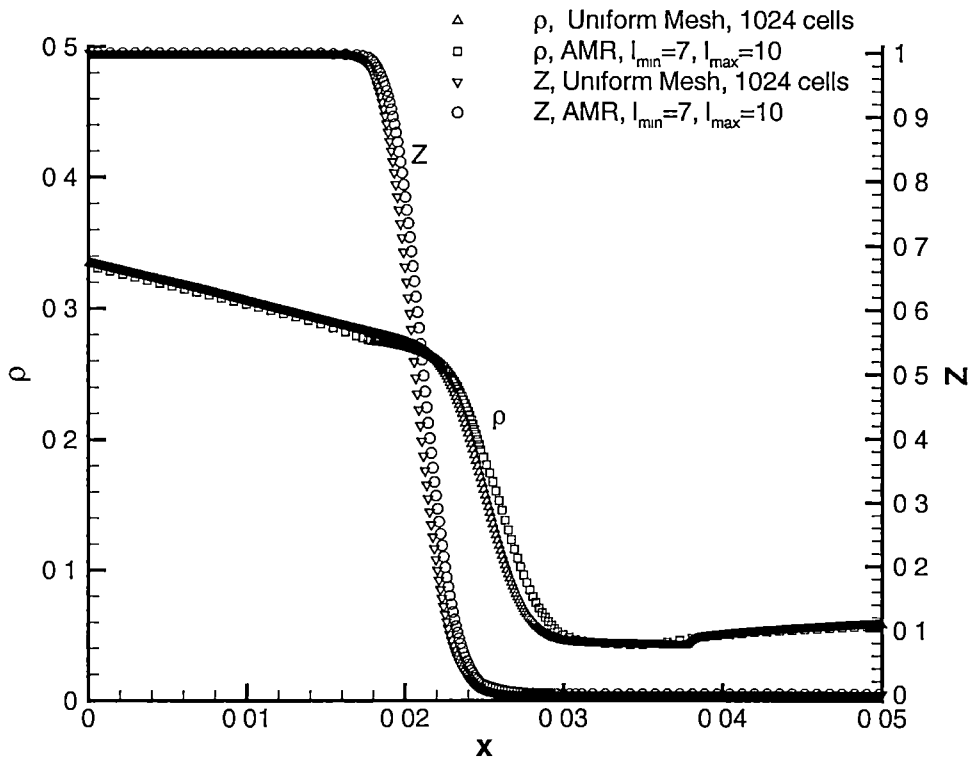


Figure 5 22 Plots of density and reactant mass fraction at time $t = 2\,0 \times 10^{-4} s$, during the filling phase for a uniform grid and AMR grid

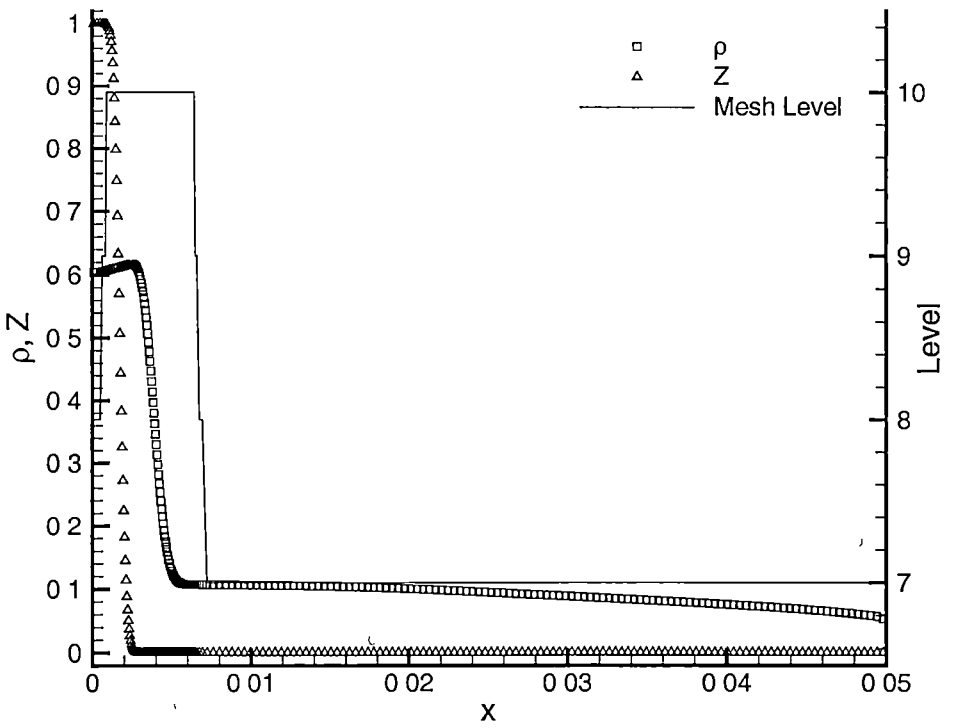


Figure 5 23 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,3 \times 10^{-4} s$, during the filling phase, using a tracking variable for the interface $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$, $\Delta x_{if} = 0\,005$

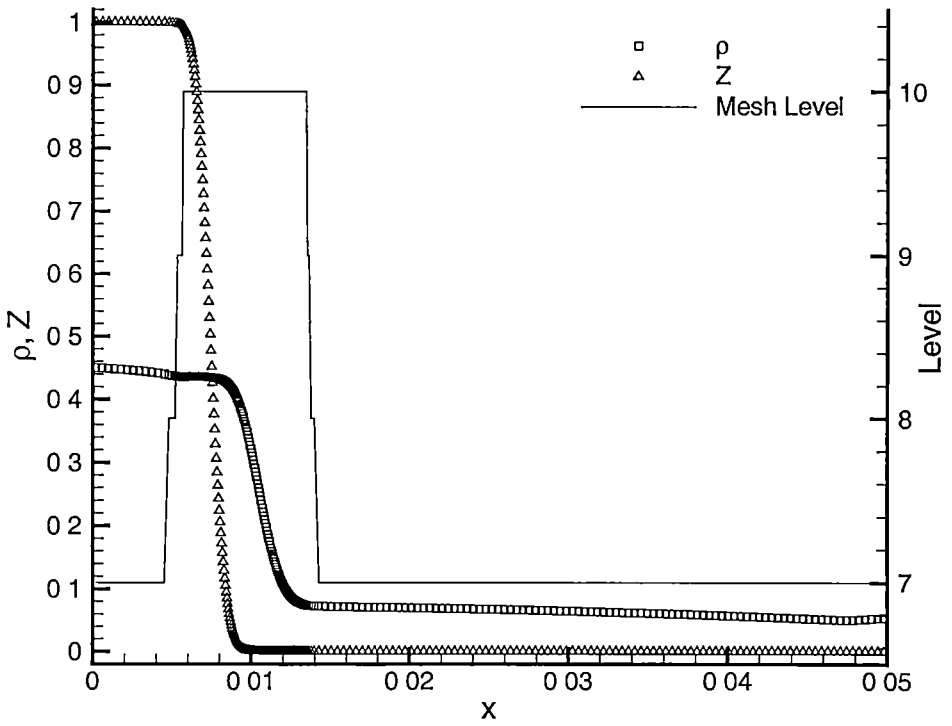


Figure 5 24 Plots of density, reactant mass fraction and mesh structure, at time $t = 1\,6 \times 10^{-4} s$, during the filling phase, using a tracking variable for the interface $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$, $\Delta x_{if} = 0\,005$

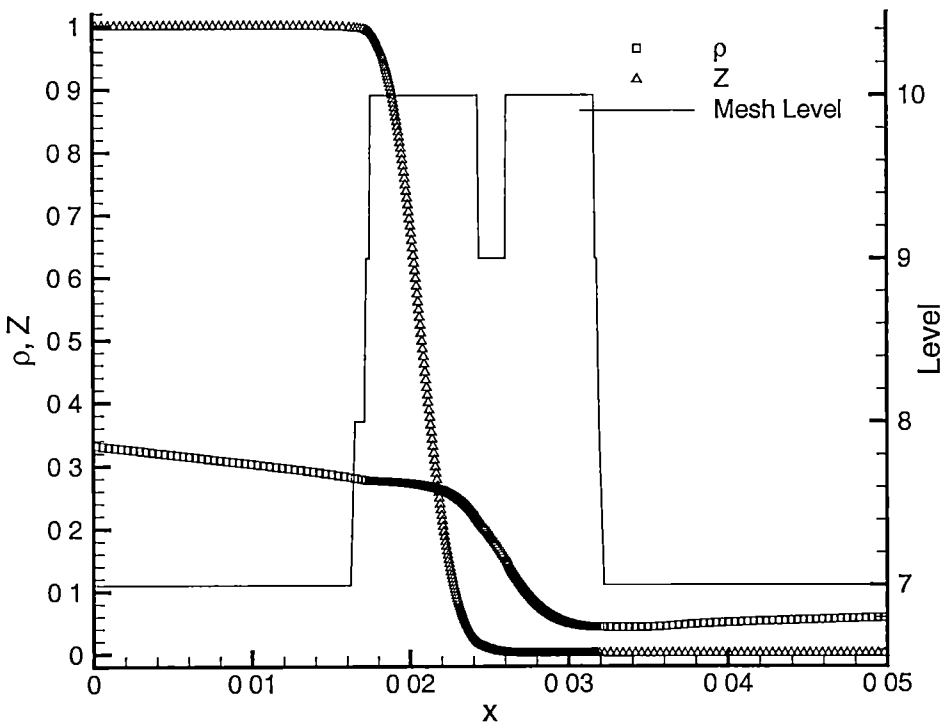


Figure 5 25 Plots of density, reactant mass fraction and mesh structure, at time $t = 2\,0 \times 10^{-4} s$, during the filling phase, using a tracking variable for the interface $l_{min} = 7$, $l_{max} = 10$, $\xi_{join} = 0\,1$, $\xi_{split} = 0\,4$, $\varepsilon^s = 1\,0$, $\varepsilon^c = 0\,05$, $\Delta x_{if} = 0\,005$

by this method A band of finely refined cells follows the interface as it moves, but no reasonable value of Δx_{if} will be large enough to contain the interface as it starts to diffuse A comparison of the interface profiles obtained using the tracking variable with those obtained using suitable values for ε^c are presented in figures 5 26 and 5 27 The results presented in this section demonstrate that the adaptive mesh refinement is able to reach the levels of accuracy obtained using a uniform mesh of comparable size with a significant cost advantage The following section discusses about techniques other than increasing the spatial resolution, for accurately resolving a contact discontinuity

5.4 Improved methods for resolving a contact discontinuity

Increasing the spatial resolution helps to capture a contact discontinuity with improved accuracy Adaptive mesh refinement is a convenient way of increasing the spatial resolution to very fine levels with insignificant increases in cost However the fact remains that, inspite of increasing the spatial resolution the contact surfaces will still remain diffused over a large number of cells This is due to the innate numerical diffusion of the numerical schemes By nature a shock wave always tends to steepen, because in any finite compression wave, the $(u + a)$ characteristics progressively approach each other coalescing into a shock wave However the contact surfaces do not have this property and they tend to diffuse The problem in numerical simulations is that, due to the presence of high numerical diffusion, the contact surfaces diffuse to levels far higher than those occuring physically The diffusion of the contact surfaces can be countered by adding certain artificial dissipation terms that behave as negative diffusion and help in

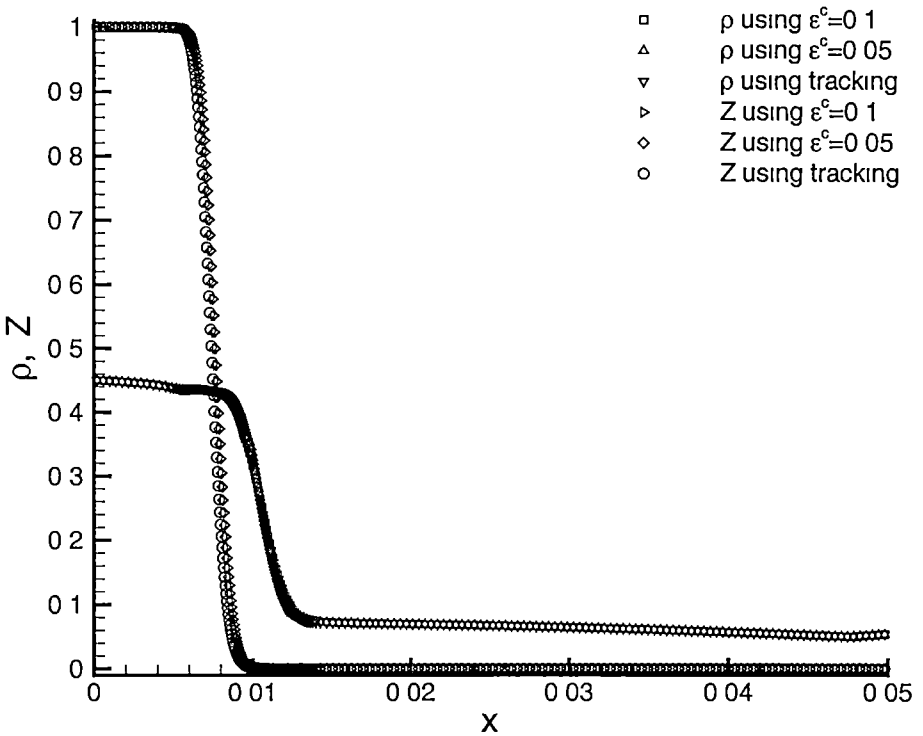


Figure 5 26 Comparison of density and reactant mass fraction profiles, with and without tracking at time $t = 1\ 6 \times 10^{-4} s$

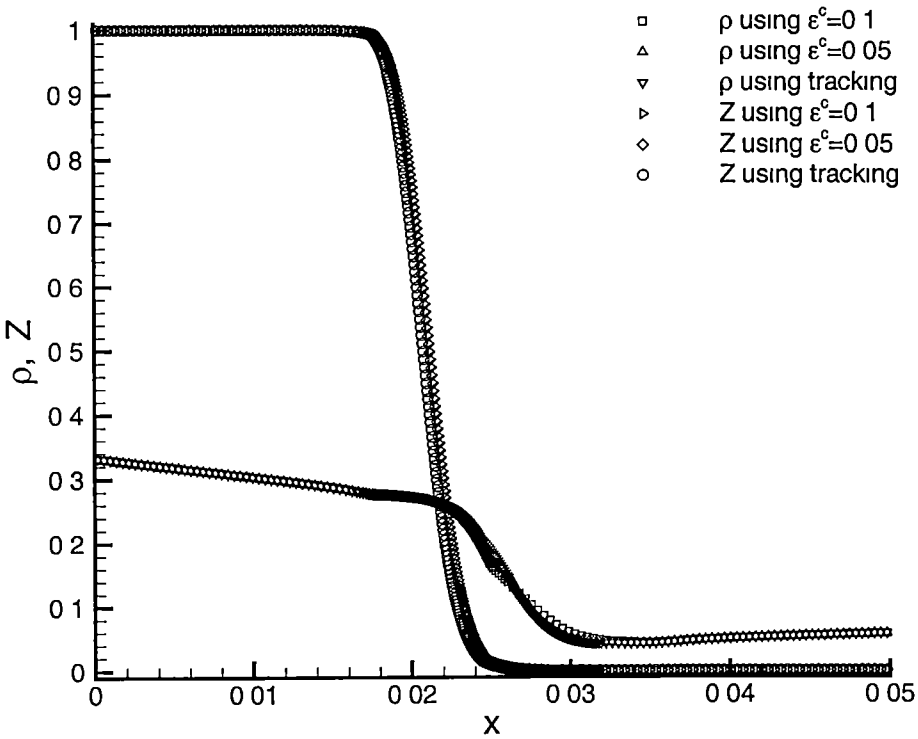


Figure 5 27 Comparison of density and reactant mass fraction profiles, with and without tracking at time $t = 2\,0 \times 10^{-4} s$

confining an interface within a few cell distances The rest of this section discusses one such technique, called the Confinement method

5.4.1 Confinement of a passive scalar discontinuity

Pulses and step discontinuities concentrated over only two or three grid cells can be convected over long distances without spreading (even on a coarse grid and with only first order finite-difference schemes), by the addition of a non-linear confinement term [14, 11] The idea of confining contact discontinuities was apparently first discussed by Harten [3] This method involves treating the features as solitary waves, which involve functions that obey a non-linear, discrete evolution equation, such that their internal structure remains fixed and confined to a thin region a few grid cells wide Basic expressions and results for one dimensional scalar advection equation have been presented by Steinhoff [14] and Puskos [11] Consider the one dimensional advection equation for a scalar w,

$$\frac{\partial w}{\partial t} = -a\frac{\partial w}{\partial x} \tag{5 3}$$

A first order upwind discretization of this equation will be

$$w_i^{n+1} = w_i^n - \sigma\left(w_i^n - w_{i-1}^n\right) \tag{5 4}$$

where n labels the time step, i the grid index and $\sigma = a\Delta t/\Delta x$ This scheme of discretization is highly diffusive A Taylor's series expansion of equation 5 4 shows that the

diffusion can be represented as,

$$\sim \mu_0 \Delta x^2 \partial_x^2 w, \text{ where, } \mu_0 = \frac{\sigma(1-\sigma)}{2}$$

The basis of the Confinement procedure is to counter this numerical diffusion by adding a term to equation 5 4 which would convect a step discontinuity inward, along its own derivative

$$w_i^{n+1} = w_i^n - \sigma\left(w_i^n - w_{i-1}^n\right) - \varepsilon f\left(w_i\right) \tag{5 5}$$

Here, εf is the "confinement term" added to the basic numerical scheme and ε is the "confinement parameter" One form of this confinement term which conserves the scalar w, maintains the actual speed of advection a and is suitable for the advection of a step discontinuity is,

$$f\left(w_i\right) = \delta_i^- \left(\frac{\left(\delta_i^- w_i^n\right)\left(\delta_i^- w_{i+1}^n\right)}{w_{i+1}^n - w_{i-1}^n} \right) \tag{5 6}$$

The operator δ_i^- is defined as $\delta_i^- f_i = f_i - f_{i-1}$ The results presented in Ref [11] show that this equation is able to confine and advect a step discontinuity in a scalar equation However this choice of the confinement term has some limitations, there exists a maximum stable value of ε for a given σ This condition reduces the effectiveness of the Confinement method when low values of σ are involved The next subsection discusses the extension of this Confinement procedure to a system of equations

5.4.2 Confinement of contact discontinuities in vector systems

The suitability of the Confinement method for the present problem depends on how best it could be extended to a vector system of equations This subsection descibes the extension of this method to Euler equations using the characteristic equations as basis The one-dimensional reactive Euler equations 2 1 can be diagonalized by pre-multiplying with $\widehat{M}^{-1}$ (refer Appendix A) to obtain the characteristic form of those equations

$$\frac{\partial \widehat{Q}\mathcal{A}}{\partial t} + \Lambda \frac{\partial \widehat{Q}\mathcal{A}}{\partial x} = \widehat{H}\mathcal{A} \tag{5 7}$$

where, $\Lambda = \mathrm{diag}\,(u, u+a, u-a, u)$ Applying the definition $\delta\widehat{Q} = \widehat{M}^{-1}\delta Q$, the following definitions of the characteristic variables $\delta\widehat{Q} = (\delta\hat{q}_1, \delta\hat{q}_2, \delta\hat{q}_3, \delta\hat{q}_4)^T$ are obtained, with $\delta\widehat{Q}$ representing an arbitrary variation, either $\dfrac{\partial}{\partial t}$ or $\dfrac{\partial}{\partial x}$

$$\begin{aligned}
\delta\hat{q}_1 &= \delta\rho - \frac{1}{a^2}\delta p \\
\delta\hat{q}_2 &= \frac{1}{2a^2}\delta p + \frac{\rho}{2a}\delta u \\
\delta\hat{q}_3 &= \frac{1}{2a^2}\delta p - \frac{\rho}{2a}\delta u \\
\delta\hat{q}_4 &= \delta\,(\rho Z) - \frac{1}{a^2}\delta p
\end{aligned} \tag{5 8}$$

From the eigenvalues in the diagonal matrix Λ, we see that two of these variables, $\hat{q}_1$ and $\hat{q}_4$ are convected at the local fluid velocity u These two variables, $\hat{q}_1$ and $\hat{q}_4$, are the characteristic variables of interest for the present problem, the variables that correspond to the ones that need to be confined for capturing interfaces within a few cell distances A confinement term can be added to the first and last characteristic equations (the

equations being convected at speed u), similar to the scalar equation 5 5

$$\frac{\partial \widehat{Q}\mathcal{A}}{\partial t} + \Lambda \frac{\partial \widehat{Q}\mathcal{A}}{\partial x} = \widehat{H}\mathcal{A} - \varepsilon \widehat{F}\mathcal{A} \tag{5 9}$$

where,

$$\widehat{F} = \left(f\left(\hat{q}_1\right), 0, 0, f\left(\hat{q}_4\right)\right)^T$$

The function f is defined in equation 5 6 Now pre-multiplying equation 5 9 by $\widehat{M}$ takes it back to the conservative form,

$$\frac{\partial Q\mathcal{A}}{\partial t} + \frac{\partial E\mathcal{A}}{\partial x} = H\mathcal{A} - \varepsilon F\mathcal{A} \tag{5 10}$$

where,

$$F = \widehat{M}\widehat{F} = \begin{pmatrix} f\left(\hat{q}_1\right) \\ u f\left(\hat{q}_1\right) \\ \frac{u^2}{2} f\left(\hat{q}_1\right) + q_0 f\left(\hat{q}_4\right) \\ f\left(\hat{q}_4\right) \end{pmatrix} \tag{5 11}$$

The modified finite volume update equation (refer equation 2 4) for the unknown vector Q is,

$$\Delta Q_i = -\frac{\Delta t}{V_i}\left[\widetilde{E}_{i+1/2} A_{i+1/2} - \widetilde{E}_{i-1/2} A_{i-1/2}\right\} + \Delta t\, H_i - \varepsilon \Delta t\, F_i \tag{5 12}$$

$$Q_i^{n+1} = Q_i^n + \Delta Q_i \tag{5 13}$$

This equation is tried on a simple problem, two interfaces moving at a constant speed in a straight tube A tube of length $L = 20m$ with both ends open is filled with the

same working fluid as the PDE model The initial condition can be stated as,

$$\begin{pmatrix} p \\ u \\ T \\ Z \end{pmatrix} = \begin{cases} \begin{pmatrix} 1\,01325 N/m^2 \\ 250 m/s \\ 400 K \\ 1 \end{pmatrix} & \text{if } 2 < x < 6 \\ \begin{pmatrix} 1\,01325 N/m^2 \\ 250 m/s \\ 300 K \\ 0 \end{pmatrix} & \text{otherwise} \end{cases} \tag{5 14}$$

The contact surfaces are at $x = 2$ and $x = 6$ and are to be convected at the constant speed u The results shown in figures 5 28 and 5 29 show that it is possible to capture the interface within fewer cell distances when a sufficiently high value of the confinement parameter ε is used However this improvement could not be repeated in more complicated flow situations like the filling phase of the PDE model This may be attributed to several reasons On important reason being that the confinement term in equation 5 6 is neither the only choice nor the best In this work, the confinement term used for a scalar advection problem was extended in a straight forward way to a system of equations No attempt was made to try using other possible expressions as confinement terms The present confinement term was found to be unstable, leading to oscillations within a few time steps It could be possible to devise a stable expression for the confinement term

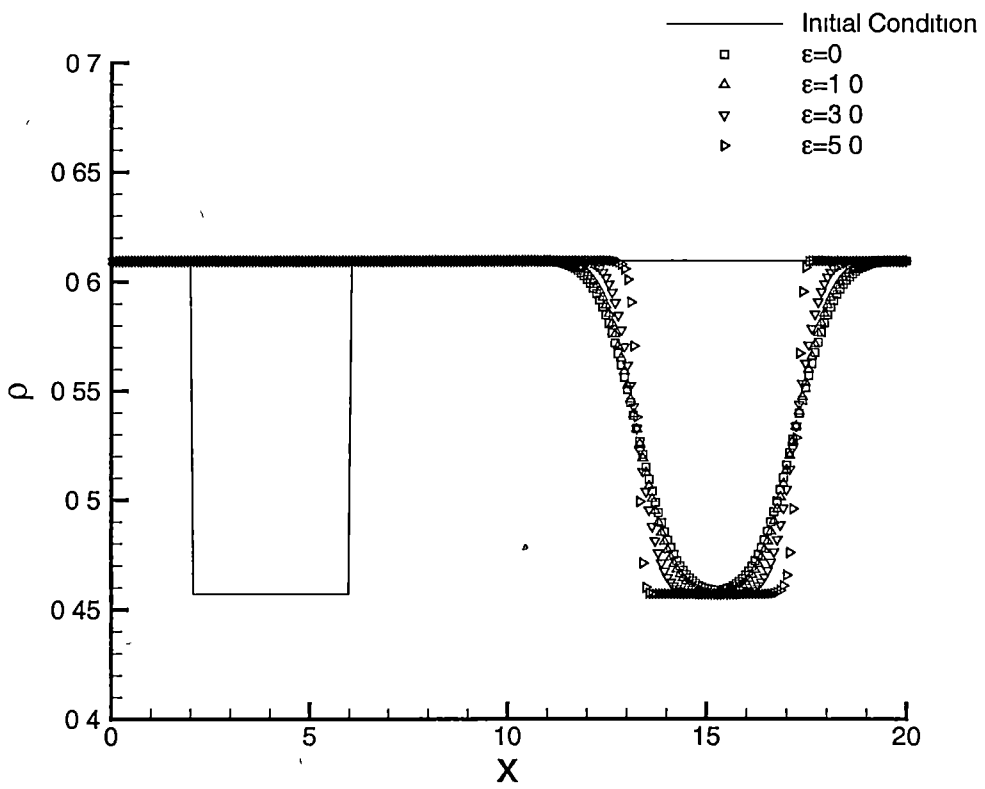


Figure 5 28 Advection of two contact surfaces with Confinement. $\sigma = 0\,9$ Results shown at time $t = 4\,5 \times 10^{-2} s$

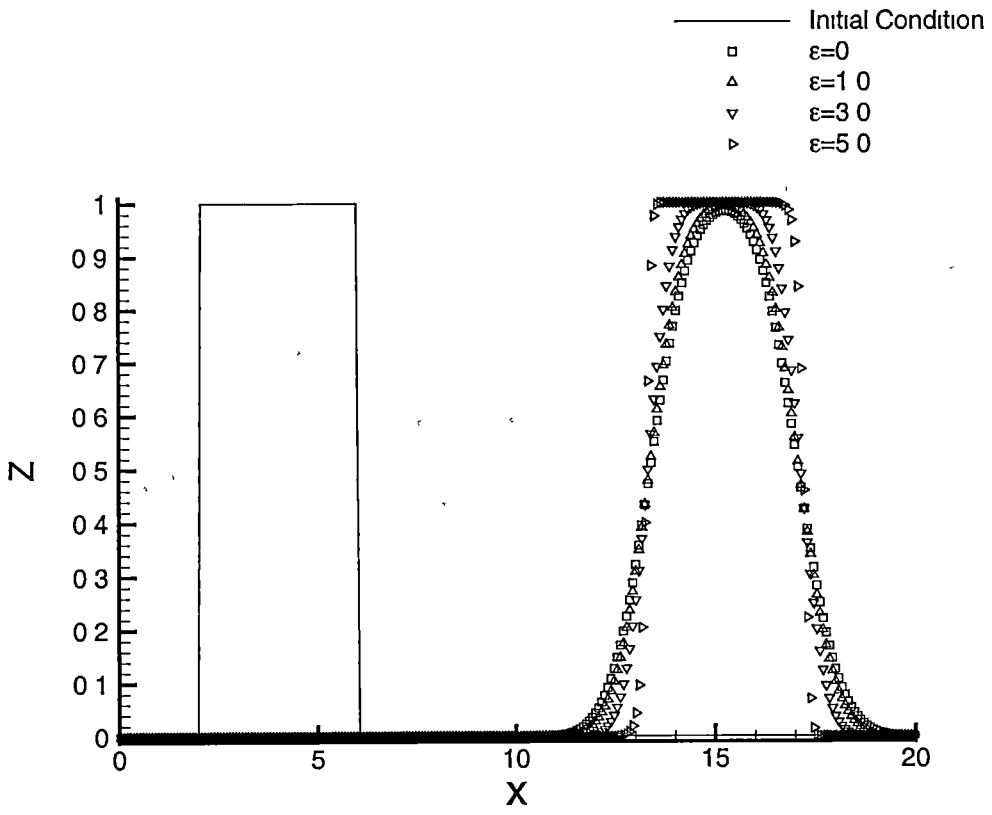


Figure 5 29 Advection of two Z-interfaces with Confinement $\sigma = 0\,9$. Results shown at time $t = 4.5 \times 10^{-2} s$

Even with such an expression, the possible difficulties are foreseeable

As is seen from the results, a higher value of the confinement parameter ε is necessary for effective Confinement The system has to be stable for large values of ε even for low values of the CFL number σ In a problem situation, such as the PDE model, the effective local CFL number based on u could easily be as low as 0 05—0 1, even when the global CFL number based on $u + a$ is set to 0 9 While negative diffusion should be added near the contact surface, care should be taken not to meddle with the diffusion near the shocks (The density and entropy are discontinuous near a shock too) When these and other concerns are addressed, Confinement of the contact surfaces should be achievable even in complicated flow situations

Chapter 6

Conclusions and Recommendations

An unsteady flow solver code was developed from scratch and used for simulating the various phases of a PDE working cycle. Simulations were carried out using two numerical schemes, Roe's approximate Riemann Solver and advection upstream splitting method ($AUSM^+$) The results show that $AUSM^+$ scheme performs as well as Roe's scheme and costs less In typical computations $AUSM^+$ consumed 50% less CPU time compared to Roe's scheme Nevertheless it still remained more costly than Chang's space-time method The effect of mesh size on the initiation and propagation of a detonation wave and the interface were studied The study showed the importance and effects of increasing the spatial resolution

A fully threaded tree (FTT) structure was used for adaptive refinement of the mesh Results obtained using this adaptive mesh refinement procedure show that it is able to reach the levels of accuracy obtained using a uniform mesh of comparable size In terms of CPU time consumed, the adaptive refinement helped in reducing the cost of computation by a factor of 10 This significant savings in cost of computation makes it

possible to increase the spatial resolution to very fine levels without increasing the cost much Even with the increase in spatial resolution, the accuracy of contact surfaces is not satisfactory This is due to the high numerical diffusion of the flux schemes

An improved method for accurately resolving the contact discontinuity by adding a suitable negative diffusion, namely the Confinement method, was tested for a simple flow problem This method yielded good results for this simple test case However it failed in complicated flow situations, like the present PDE model

Future work can be directed towards extending the flow solver to multi-dimensions Also a multi-species full kinetic model can be implemented to model the chemical reaction instead of the present simplified chemisty model Making the solver second-order accurate could be helpful in capturing the contact discontinuities more accurately However Confinement seems to be the best option for capturing the contact discontinuities with improved accuracy Some of the difficulties that might arise in the extension of the Confinement method to complicated flow situations have been identified The most notable among them being the presence of multiple CFL numbers in the flow field More attention needs to be paid towards developing better expressions for the confinement term which would remain stable and effective over a broad range of CFL numbers The Confinement method should also be tailored such that it adds negative diffusion only near the interfaces without meddling with the shocks and detonation fronts

Bibliography

Bibliography

[1] Cambier, J L , " Preliminary Modeling of Pulse Detonation Rocket Engines", AIAA, Proceedings of 35th JPC, 1999,2659

[2] Chang, S C , "The method of Space-Time Conservation Element and Solution Element — A new approach for solving the Navier-Stokes and Euler Equations", Journal of Computational Physics(1995), 119, pp 295–324

[3] Harten, A , "The artificial compression method for computation of shocks and contact discontinuities III Self-adjusting hybrid schemes", Mathematics of Computation, Vol 32, No 142 April 1978

[4] Harten, A and Hyman, J M , "Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws", Journal of Computational Physics(1983), 50, pp 235–269

[5] Hirsch, C , "Numerical Computation of Internal and External Flows", Vol 1&2, John Wiley & Sons

[6] Khokhlov, A M , "Fully Threaded Tree Algorithms for Adaptive Refinement Fluid Dynamics Simulations", Journal of Computational Physics(1998), 143, pp 519–543

[7] Liou, M S , " A New Flux Splitting Scheme", Journal of Computational Physics(1993), 107, pp 23–39

[8] Liou, M S , "Progress towards an improved CFD Method AUSM", AIAA(1995), 1701, pp 606–625

[9] Liou, M S , "AUSM schemes and extensions for low mach and multiphase flows", VKI Lecture Series(1999)-03

[10] Mohanraj, R and Merkle, C L , "A numerical study of Pulse Detonation Engine performance", AIAA — Proceedings of 38th Aerospace Sciences Meeting and Exhibit(2000), 0315

[11] Puskas, E G , "1-D Confinement — Master's Thesis", University of Tennessee Space Institute(1996)

[12] Quirk, J , " Godunov-type Schemes applied to Detonation Flows", ICASE Report, 1993, 93-15

[13] Roe, P L , "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", Journal of Computational Physics(1981), 43, pp 357–372

[14] Steinhoff, J et al , "Computation of thin features over long distances using Solitary waves", AIAA paper, 1997

[15] Turns, S R , "An Introduction to Combustion — Second Edition", McGraw Hill(1999)

Appendix

Appendix A

Jacobians and Eigensystem for Reactive Euler Equations

The conservative form of the quasi-one-dimensional reactive Euler equations can be written as,

$$\frac{\partial QA}{\partial t} + \frac{\partial EA}{\partial x} = HA \tag{A 1}$$

where $A = A(x)$ is the area, Q is the unknown vector, E is the flux vector and H is the source term

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho e \\ \rho Z \end{pmatrix}, \quad E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (\rho e + p)\, u \\ \rho u Z \end{pmatrix}, \quad H = \begin{pmatrix} 0 \\ \dfrac{p}{A}\dfrac{dA}{dx} \\ 0 \\ -K\rho Z \exp^{-E_a/R_u T} \end{pmatrix} \tag{A 2}$$

The flux Jacobian $A = \dfrac{\partial E}{\partial Q}$ can be written as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{(\gamma-3)\,u^2}{2} & -(\gamma-3)\,u & (\gamma-1) & -(\gamma-1)\,q_0 \\ \begin{pmatrix} (\gamma-1)\,u^3 - \gamma e u \\ +(\gamma-1)\,q_0 u Z \end{pmatrix} & \begin{pmatrix} -\dfrac{3}{2}(\gamma-1)\,u^2 + \gamma e \\ -(\gamma-1)\,q_0 Z \end{pmatrix} & \gamma u & -(\gamma-1)\,q_0 u \\ -uZ & Z & 0 & u \end{bmatrix} \tag{A 3}$$

Equation A 1 can be premultiplied by $\widehat{M}^{-1} = \dfrac{\partial \widehat{Q}}{\partial Q}$ ($\widehat{M}$ is the matrix of right eigenvectors of A) to obtain the characteristic form,

$$\frac{\partial \widehat{Q}A}{\partial t} + \Lambda \frac{\partial \widehat{Q}A}{\partial x} = \widehat{H}A \tag{A 4}$$

which introduces the diagonal matrix of eigenvalues,

$$\Lambda = \begin{bmatrix} u & & & \\ & u+a & & \\ & & u-a & \\ & & & u \end{bmatrix} \tag{A 5}$$

The transformation matrices are,

$$\widehat{M}^{-1} = \begin{bmatrix} 1 - \dfrac{(\gamma-1)M^2}{2} & \dfrac{(\gamma-1)M}{a} & -\dfrac{(\gamma-1)}{a^2} & \dfrac{(\gamma-1)q_0}{a^2} \\ -\dfrac{M}{2}\left(1 - \dfrac{(\gamma-1)M}{2}\right) & \dfrac{1-(\gamma-1)M}{2a} & \dfrac{(\gamma-1)}{2a^2} & -\dfrac{(\gamma-1)q_0}{2a^2} \\ \dfrac{M}{2}\left(1 + \dfrac{(\gamma-1)M}{2}\right) & \dfrac{-1-(\gamma-1)M}{2a} & \dfrac{(\gamma-1)}{2a^2} & -\dfrac{(\gamma-1)q_0}{2a^2} \\ -\dfrac{(\gamma-1)M^2 Z}{2} & \dfrac{(\gamma-1)MZ}{a} & -\dfrac{(\gamma-1)Z}{a^2} & 1 + \dfrac{(\gamma-1)q_0 Z}{a^2} \end{bmatrix} \tag{A 6}$$

$$\widehat{M} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ u & u+a & u-a & 0 \\ \dfrac{u^2}{2} & \dfrac{u^2}{2} + ua + \dfrac{a^2}{(\gamma-1)} + q_0 Z & \dfrac{u^2}{2} - ua + \dfrac{a^2}{(\gamma-1)} + q_0 Z & q_0 \\ 0 & Z & Z & 1 \end{bmatrix} \tag{A 7}$$

Here M is the mach number and a the speed of sound

Vita

Ramanan Sankaran was born in Madurai, India, on the 29th of January 1977. He lived the first 21 years of his life in the hot humid weather of Chennai (erstwhile Madras), India This is where he did his schooling and undergraduate studies He joined Anna University, Chennai in 1994 After spending 4 years in the midst of trees and deer in the wonderful Guindy campus, he earned his Bachelor's degree in mechanical engineering in 1998 He came to Tennessee in August 1998 After spending another 2 years among deer at UTSI, he has now completed his degree requirements for the Master of Science degree in mechanical engineering He is shifting further north to colder climates He is planning to pursue his Ph D, once again in mechanical engineering, at the University of Michigan