AN INVESTIGATION OF FLUX-SPLITTING ALGORITHMS FOR CHEMICALLY REACTING FLOWS

By

Rajasekhar Venkata Darapuram

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computational Engineering
in the Department of Computational Engineering

Mississippi State, Mississippi

May 2001

AN INVESTIGATION OF FLUX-SPLITTING ALGORITHMS FOR CHEMICALLY REACTING FLOWS

By

Rajasekhar Venkata Darapuram

Approved:

Pasquale Cinnella
Associate Professor of Aerospace Engineering
(Director of Dissertation)

Edward Luke
Assistant Professor of Computational Engineering
(Committee Member)

Bharat K. Soni
Professor of Aerospace Engineering
(Committee Member)

Roy Koomullil
Research Engineer of Computational Engineering
(Committee Member)

Bradley Carter
Professor and Graduate Coordinator of the Department of Computational Engineering

Wayne A. Bennett
Dean of the College of Engineering

Name: Rajasekhar Venkata Darapuram

Date of Degree: May 12, 2001

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. Pasquale Cinnella

Title of Study: AN INVESTIGATION OF FLUX-SPLITTING ALGORITMS FOR CHEMICALLY REACTING FLOWS

Pages in Study: 122

Candidate for Degree of Master of Science

This paper presents an investigation of seven different flux splitting algorithms for the discretization of inviscid fluxes, which are the primary source for the non-linear behavior (eg. shocks, contact discontinuities). The aim of the present work is to enhance the accuracy and robustness of CHEM, a three-dimensional flow solver, which is capable of simulating a wide range of flow conditions, including chemical non-equilibrium. Five different test cases cases are considered and thoroughly analyzed. The overall goal is to find a numerical scheme that can meet some stringent specifications of efficiency, accuracy and robustness on the widest possible spectrum of flow conditions.

DEDICATION

I would like to dedicate this research to my sister Annapurna, and my parents.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major professor, Dr. Pasquale Cinnella, for his guidance and continuing assistance throughout this work. I would like to offer a special "thank you" to Dr. Edward Luke, who has significantly contributed to my learning experience during my graduate studies. I would also like to thank the rest of my committee, Dr. Bharat Soni and Dr. Roy Koomullil for their contributions. In addition I would like to thank Dr. Donald Trotter and the Engineering Research Center for providing financial support and facilities to this research effort.

I would like to thank my friends Amithkumar Reddy and PhaniKant Mantena for always supporting and encouraging me. Also, I would like to thank Dr. Xiao Ling-Tong and Dr. Junxiao Wu for their help and guidance.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

NOMENCLATURE

Symbol	
a	Speed of sound
CFL	Courant-Friedrichs-Lewy number
c_{v_s}	Species specific heat at constant volume
c_{p_s}	Species specific heat at constant pressure
D	Diffusion coefficient, Main diagonal matrix
e_0	Energy, Total energy per unit mass
e_s	Species internal energy
F, G, H	Inviscid flux vectors in generalized coordinates
h_{f_s}	Species heat of formation
h_0	Total enthalpy
i,j,k	Unit vectors for a cartesian system
I	Identity matrix
J	Jacobian of the inverse metric transformation
k	Thermal conductivity
$K_{f,r}$	Forward reaction rate
$K_{b,r}$	Backward reaction rate
$K_{c,r}$	Reaction equilibrium constant for a chemical reaction

L	Lower diagonal matrix
L_e	Lewis number
l,r	Left and right states
$\dot{m}$	Mass flow
M	Mach number
$\hat{M}$	Species molecular weight
n	Time level index
$\tilde{n}$	Unit vector normal to a boundary surface
NS	Number of species in the fluid
p	Static pressure
P_r	Prandtl number
$\tilde{q}$	Heat flux
Q	Dependent variable vector in cartesian coordinates
$\hat{Q}$	Dependent variable vector in generalized coordinates
R	Ideal gas constant, Numerical residual
R_s	Species gas constant
s_0	Reference entropy
t	Time in physical space
T	Fluid temperature
$\tilde{u}$	Contravariant velocity component

u,v,w	Cartesian components of velocity vector
U	Upper diagonal matrix
V_D	Diffusion velocity
X_s	Chemical species
W	Vector of source terms
x,y,z	Cartesian Axes
$\gamma, \tilde{\gamma}$	Ratio of specific heats, Mixture ratio of frozen specific heats
$\Omega, \partial\Omega$	Control volume, Control surface
λ	Eigenvalue of the flux Jacobian matrix, Second coefficient of viscosity
μ	Dynamic viscosity
v', v''	Stoichiometric coefficients
ξ, η, ζ	Generalized Coordinates
$\dot{\omega}_s$	Species Production rate
$\bar{\bar{\tau}}$	Fluid Stress Tensor
θ_c	Cell volume
ρ	Global density
ρ_s	Species density
∂_{xy}	Kronecker's delta function

$\theta_{v,s}$	Species characteristic with temperature

Operators

$\rightarrow$	Mapping (composition) operator
$a \leftarrow b$	rule operator,(meaning *b* generates *a*)
$\oplus$	A generic operator
$\in$	Belongs to the set
$\langle \ \rangle$	Arithmetic average
$\| \ \|$	Jump operator

CHAPTER I

INTRODUCTION

The long-term objective of the present endeavor is the development of a flow solver capable of accurate simulations of flow fields in the entire Mach number range. The need for a simulation tool optimized over a range of Mach numbers as wide as possible emerges in many applications, such as aero-propulsive systems, where high speed flow fields may contain re-circulation zones with very low speeds. High Mach numbers and high temperatures lead to chemical reactions and dissociation that can make the perfect gas model unsuitable. Furthermore, even in the case of totally subsonic flows, the incompressible approach can be precluded by the presence of chemical reactions and heat transfer, (which happens in simulations of combustion problems, for example). An additional motivation for a unified approach to the numerical formulation comes from the increasing demand of cost reduction and productivity of the "*Research and Development*" sector of the industry.

The key issue in the present study was to analyze the discretization of the inviscid fluxes, which were the primary source for the non-linear behavior (shocks, contact discontinuities). Presently, several upwind schemes have been developed in the scientific literature. Most of them are usually categorized as either Flux Difference Splitting (FDS) or Flux Vector Splitting (FVS). The former normally uses an approximate solution of a local Riemann problem. The most successful and popular approximate Riemann solver is

Roe's method, which is the exact solution of a linearized Riemann problem[1]. It is able to capture any single stationary discontinuities with no numerical dissipation. Unfortunately non-physical solutions such as expansion shocks can be generated by this scheme, although they can be removed by using an "*Entropy Fix*[2]". At strong expansions, however, the Roe scheme tends to diverge even if the entropy fix is applied. In addition, it is known that the Roe scheme has a more serious problem, the so called "*carbuncle phenomenon*", which is a numerical instability that arises when capturing strong shock waves in multidimensional computations. Recently Quirk[3] has reported that even exact Riemann solvers suffer from this problem. Roe's approximate Riemann solver has consistency, and this type of approximate Riemann solvers are called "*Godunov-type*" schemes[4]. Another Godunov-type scheme is the HLLE scheme, which simplifies the Riemann problem by expressing the solution with only two waves. Einfeldt[5, 6] has shown that this scheme satisfies the positivity and entropy conditions if some appropriate velocities are chosen for these two waves. However, the HLLE scheme lacks the information on contact discontinuities, so it is too dissipative to be applied to the Navier-Stokes equations, although the numerical dissipation is strong enough to remove the carbuncle phenomenon[3]. The HLLE scheme can be modified to include the information on the contact discontinuity under the positivity condition[7], however, that scheme again suffers from carbuncle phenomena.

Other way of introducing upwinding is FVS, in which the flux function is divided into positive and negative parts, which give the numerical flux at a cell interface. The idea is to reproduce the "*correct*" direction of propagation of information for a hyperbolic system of partial differential equations. Steger and Warming[8] made use of similarity transformations and the homogeneity property of the Euler equations, splitting the flux depending on the sign of the eigenvalues of the flux Jacobian matrix. However, this splitting showed errors around the sonic point, and Van Leer[9] proposed an alternative splitting, which gives noticeably better results and produces steady shock profiles. Hanel[10] further modified Van Leer's FVS so as to preserve total enthalpy for the steady state solution. These FVS schemes, however, share a serious disadvantage: large dissipation on contact discontinuities and shear layers.

A more recent effort to develop less-dissipative upwind schemes focuses on reducing the surplus dissipation of the FVS by introducing the flavor of FDS into FVS schemes. Liou and Steffen proposed a promising scheme named Advection Upstream Splitting Method (AUSM)[11, 12], in which the cell-face advection Mach number is appropriately defined to determine the upwind extrapolation for the convective quantities. The AUSM scheme can capture a stationary contact discontinuity with no numerical dissipation, and is robust enough to calculate strong shock waves. However, it creates a slight numerical overshoot immediately behind the shock. Other variants, termed AUSMD and AUSMV, have been developed by Wada and Liou[13] to achieve the following features: accurate resolution of stationary and moving shock and contact discontinuities , and positivity preserving of pressure and density[14]. The AUSMDV, a blending of AUSM, flux-difference and flux-vector splittings, improves the robustness of AUSM in dealing with the collision of strong shocks. However, the "*carbuncle phenomenon*" appears, albeit much weaker than the one resulting from the Roe scheme, which requires a fix. The AUSM$^+$[15] is designed to be a further improvement of AUSM, and has the following features in addition to the earlier features of the AUSM scheme:

improvement in accuracy over its predecessor (AUSM) and other popular schemes, simplicity, and easy generalization to other conservation laws. The Low Speed AUSM$^+$ scheme[16] proposes the concept of "*numerical speed of sound*" in the construction of the numerical flux. It was shown that this variable was responsible for the accurate resolution of discontinuities, such as contacts and shocks. As a result, the numerical dissipation for low speed flows is scaled with the local fluid speed, rather than the sound speed. Hence, accuracy is enhanced, more correct solutions at low speed are recovered, and the convergence rate is improved.

The present work constitutes a preliminary step necessary to build the basis for future developments. A particular subject on which the attention has been focused is the comparative investigation of eight different algorithms for the discretization of the inviscid fluxes in the Euler equations (Roe, Steger-Warming, Van Leer, AUSMD, AUSMV, AUSMDV, AUSM$^+$ and Low Speed AUSM$^+$).

In the following, Chapter II reviews briefly the governing equations for the general case of a flow field featuring mixtures of gases in chemical non-equilibrium. Moreover, the chapter introduces curvilinear coordinate transformations of the above equations and viscous fluxes. The finite volume approach and eigenvalues and eigenvectors are also briefly discussed.

The thermodynamic models, caloric equation of state and the treatment of finite rate chemistry are presented in Chapter III. A brief introduction to transport properties is also discussed. Chapter IV deals with the Loci framework, in which the chemically reacting flow solver was developed.

The details of the spatial and temporal discretizations of the governing equations by means of the finite volume approach are presented in Chapter V, which also contains a brief description of the extrapolation of the primitive variables to the faces. Solution techniques for the (large) resulting linear system of algebraic equations are also presented.

Chapter VI discusses the different flux-splitting techniques: the Roe flux-difference splitting[1], Van Leer flux-vector splitting[9], Steger Warming flux-vector splitting[8], and the AUSM family of schemes[13, 15, 16].

An extensive set of steady and unsteady numerical simulations are analyzed in Chapter VII. Different test cases encompassing a wide variety of flow conditions, both viscous and inviscid, ranging from the low subsonic to the hypersonic regimes, are utilized to evaluate and compare accuracy, efficiency, and robustness of the eight flux-splitting schemes.

Finally Chapter VIII provides a summary of the work done, and some conclusions are drawn, focusing in particular on the advantages and the disadvantages of the various schemes. Future objectives of this research are also addressed at that point.

CHAPTER II

GOVERNING EQUATIONS

This chapter describes the equations that govern the three-dimensional flow of an inviscid as well as a viscous, chemically reacting gas. The way in which a fluid interacts with its environment may be described by a system of partial differential equations, which represent the mathematical formulations of the physical laws of conservation of mass, momentum and energy. The equations used in this chapter are presented in [17]. For mathematical closure, they must be supplemented by the thermal and caloric equations of state. The equations will be first introduced in Cartesian coordinates for clarity, then extended to general curvilinear coordinates, which are used to describe problems with complicated boundaries, and finally written using a control volume approach, suitable for numerical discretization.

2.1 Equations in Cartesian Coordinates

The governing equations are an extension of the Navier-Stokes equations for a perfect gas, the main difference being that a mass conservation statement for each species must be included. That is, for a gas mixture containing *NS* distinct species, it is necessary to write NS+4 partial differential equations (NS continuity equations, three momentum equations and one energy equation). The production/disappearance of each chemical species is governed by a source term and will be discussed in detail later.

Some assumptions are made in order to reduce the complexity of the governing equations namely, no body forces and Newtonian fluid (shear stress proportional to deformation).

With the above assumptions in mind, the governing equations for a fluid flow in chemical non-equilibrium may be written in strong conservation law form for a three dimensional, time-dependent, general curvilinear coordinate system, as follows

$$\frac{\partial \mathrm{Q}}{\partial t} + \frac{\partial(\mathrm{F} - \mathrm{F_v})}{\partial x} + \frac{\partial(\mathrm{G} - \mathrm{G_v})}{\partial y} + \frac{\partial(\mathrm{H} - \mathrm{H_v})}{\partial z} = \mathrm{W}\,, \tag{2.1}$$

Where the dependent variable vector, Q, and the flux vectors F, G, and H are defined as

$$\mathrm{Q} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{NS} \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 \end{bmatrix}, \qquad \mathrm{F} = \begin{bmatrix} \rho_1 u \\ \rho_2 u \\ \vdots \\ \rho_{NS}\, u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uh_0 \end{bmatrix}, \qquad \mathrm{G} = \begin{bmatrix} \rho_1 v \\ \rho_2 v \\ \vdots \\ \rho_{NS}\, v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vh_0 \end{bmatrix},$$

$$\mathrm{H} = \begin{bmatrix} \rho_1 w \\ \rho_2 w \\ \vdots \\ \rho_{NS}\, w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wh_0 \end{bmatrix}, \qquad \mathrm{W} = \begin{bmatrix} \dot{}_1 \\ \dot{}_2 \\ \vdots \\ \dot{}_{NS} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.2}$$

F_v, G_v and H_v are the viscous flux vectors and are discussed in [18].

In the above, **W** is the vector of source terms, NS represents the number of species present in the fluid, ρ_s is the sth species density, ρ is the mixture density, h_0 is the total enthalpy, e_0 is the total energy per unit mass, P is the pressure, and *u, v* and *w* are the velocity components in the x, y and z directions, respectively. The total energy per unit volume, ρe_0 is defined as

$$\rho e_0 = \rho e + \frac{\rho q^2}{2}\,, \tag{2.3}$$

where

$$q^2 = u^2 + v^2 + w^2. \tag{2.4}$$

The consistency of species conservation with global conservation results in the following property for the $\dot{}_s$.

$$\sum_{s=1}^{NS} \dot{}_s = 0 \tag{2.5}$$

More details on viscous fluxes will be presented later in this chapter and also in chapter 3.

The pressure is calculated from Dalton's Law, which states that the pressure of a mixture of gases is the sum of the partial pressures. Each species partial pressure is obtained from the ideal gas law. This is the thermal equation of state given by

$$p = \sum_{s=1}^{NS} p_s = \sum_{s=1}^{NS} \rho_s R_s T\,, \tag{2.6}$$

where R_s is the species gas constant and ρ_s is the species density and T is the temperature. The species gas constant is computed by the equation

$$R_s = \frac{\hat{R}}{\mathrm{M}_s}\,, \tag{2.7}$$

where M_s is the species molecular mass and $\hat{R}$ is the universal gas constant. A caloric equation of state is also needed to close the mathematical problem, and will be discussed in more detail in chapter 3.

2.2 Curvilinear Coordinate Transformation

Equation 2.1 was written in Cartesian coordinates. This form of the equations is appropriate for problems where a uniform, rectangular grid can be used to model the physical geometry. In order to compute flows involving arbitrary shapes and the complex boundaries associated with these shapes, it is necessary to transform equation (2.1) to a general body-fitted curvilinear coordinate system (ξ, η, ζ). This is accomplished using the transformation equations listed below (See [19] for a derivation of this transformation):

$$\begin{aligned} &\xi = \xi(x, y, z), \\ &\eta = \eta(x, y, z), \\ &\zeta = \zeta(x, y, z), \\ &\tau = \mathrm{t}. \end{aligned} \qquad (2.8)$$

After some algebra, equation 2.1 becomes

$$\frac{\partial \hat{\mathrm{Q}}}{\partial \tau} + \frac{\partial\left(\hat{\mathrm{F}} - \hat{\mathrm{F}}_{\mathrm{v}}\right)}{\partial \xi \omega} + \frac{\partial\left(\hat{\mathrm{G}} - \hat{\mathrm{G}}_{\mathrm{v}}\right)}{\partial \eta \omega} + \omega \frac{\partial\left(\hat{\mathrm{H}} - \hat{\mathrm{H}}_{\mathrm{v}}\right)}{\partial \varsigma \omega} = \omega \hat{\mathrm{W}}\ , \qquad (2.9)$$

where $\hat{\mathrm{Q}}$ is the dependent variable factor, $\hat{F}$, $\hat{G}$, and $\hat{Q}$ are the inviscid flux vectors and $\hat{F}_v$, $\hat{G}_v$ and $\hat{H}_v$ are the viscous flux vectors. The dependent variable vector and flux vectors are now defined as follows:

$$\hat{Q} = \omega J Q\ , \qquad \hat{W} = \omega J W\ ,$$

$$\hat{F} = \tilde{\xi}_x F + \tilde{\xi}_y G + \tilde{\xi}_z H \ , \tag{2.10}$$

$$\hat{F}_v = \tilde{\xi}_x F_v + \tilde{\xi}_y G_v + \tilde{\xi}_z H_v \, .$$

Performing the necessary algebra, $\hat{F}$ becomes

$$\hat{\mathsf{F}} = J|\nabla\xi| \begin{bmatrix} \rho_1 \tilde{u} \\ \rho_2 \tilde{u} \\ \vdots \\ \rho_{NS} \tilde{u} \\ \rho u \tilde{u} + \tilde{\xi}_x P \\ \rho v \tilde{u} + \tilde{\xi}_y P \\ \rho w \tilde{u} + \tilde{\xi}_z P \\ \rho \tilde{u} h_0 \end{bmatrix}, \tag{2.11}$$

where the contravariant velocity, $\tilde{u}$ (normal to the ξ surface), is defined as:

$$\tilde{u} = \tilde{\xi}_x u + \tilde{\xi}_y v + \tilde{\xi}_z w, \tag{2.12}$$

Similarly the fluxes in the other directions can be developed using η, ζ in lieu of ξ. The normalized metrics $\tilde{\xi}_x, \tilde{\xi}_y, \tilde{\xi}_z$ in the equation (2.12) are given by

$$\tilde{\xi}_i = \frac{\xi_i}{|\nabla\xi|}, \qquad |\nabla\xi| = \sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}. \tag{2.13}$$

The Jacobian of the inverse transformation, J, and the metric quantities ξ_x, ξ_y, ξ_z can be evaluated from the transformations, and are presented in detail in [20].

2.3 Viscous Fluxes

The general expression for the stress tensor of a Newtonian fluid contains the sum of the normal and shear stress components:

$$\bar{\bar{\sigma}} = \sigma_n I + \bar{\bar{\tau}} \tag{2.14}$$

where *I* is the identity matrix, $\overline{\overline{\tau}}$ is the shear stress tensor, and $\sigma\varphi_{n}$is a scalar given by the following expression:

$$\sigma\varphi_{n} = -\varphi\omega + \left(\omega\lambda + \omega\frac{2}{3}\mu\right)\nabla\cdot\omega\left(\frac{\partial u}{\partial x} + \omega\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right). \qquad (2.15)$$

In the above, μ is the dynamic viscosity, and λ is the second coefficient of viscosity. Equation (2.15) is a function of the hydrostatic pressure p, the divergence of the velocity, and the bulk viscosity $\left(\lambda + \omega\frac{2}{3}\mu\right)$. Using Stoke's hypothesis, the bulk viscosity is set to zero (this also gives a relation for λ as a function of the dynamic viscosity μ). This assumption is acceptable as long as rotational equilibrium can be assumed, since it can be shown that the bulk viscosity is proportional to the relaxation time of the rotation energy[21]. A generic element of the shear tensor can be expressed as follows:

$$\tau\varphi_{xy} = \mu\left(\frac{\partial u}{\partial y} + \omega\frac{\partial v}{\partial x}\right) - \omega\frac{2}{3}\mu\delta_{xy}\left(\frac{\partial u}{\partial x} + \omega\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right), \qquad (2.16)$$

where $\delta\varphi_{xy}$ is the Kronecker delta, and similar expressions hold for the other terms.

The heat flux vector due to conduction in a gas in thermal equilibrium can be calculated by means of Fourier's law:

$$\overline{q} = -k\nabla T\,, \qquad (2.17)$$

where *k* is the coefficient of thermal conductivity [22].

2.4 Finite Volume Approach

The numerical solution of the fluid equations presented in the earlier sections is obtained by applying the finite volume method. The integral form of the equations is used for this purpose. This method is selected because it can guarantee that numerical truncation error does not violate conservation properties. Before introducing the finite volume formulation, it is helpful to transform the mass, momentum and energy equations into a form that is more readily usable for the numerical discretization. Green's theorem and other identities are used and the equations are recast in integral form, for an arbitrary volume(cell) *c*, as follows

$$\frac{d}{dt}\int_{\Omega_c} Q dV + \int_{\partial\Omega_c}\left(\tilde{F} - \tilde{F}_v\right) dS = \int_{\Omega_c} \dot{W} dV \, . \qquad (2.18)$$

In the above equation, Ω represents the control volume, ∂Ω is its surface. Q, $\dot{W}$ are unchanged from equation (2.2). $\tilde{F}$ represents $\hat{F}$, $\hat{G}$ or $\hat{H}$, depending on the orientation of ∂Ω. Similar is the case for F_v.

2.5 Eigenvalues and Eigenvectors

The Jacobian matrix and the eigensystem of the inviscid fluxes are important for the propagation of information and for the derivation of a scheme. From Cox [23] and Cinnella [22], the eigenvalues of the inviscid flux Jacobians for the coordinate ξ are

$$\lambda^i = J\left|\nabla\xi\right|\tilde{u}, \quad i = 1, \ldots\ldots, NS+2$$

$$\lambda^{NS+3} = J\left|\nabla\xi\right|\left(\tilde{u} + a\right), \qquad (2.19)$$

$$\lambda^{NS+4} = J\left|\nabla\xi\right|\left(\tilde{u} - a\right),$$

where *a* is frozen sound speed, to be defined in chapter 3. The eigenvalues for the coordinates η and ζ can be derived similarly.

Notice the multiplicity of the first eigenvalue. For a three-dimensional, compressible, perfect-gas formulation, there are five eigenvalues with the first eigenvalue having a multiplicity of three. For the finite-rate formulation, the multiplicity of the first eigenvalue is *NS+2*.

The right and left eigenvectors of the inviscid flux Jacobians are also obtained by following [23] and [22]. The matrix R, whose columns are the right eigenvectors, is given by

$$R = \omega \begin{bmatrix} \frac{\rho_1}{\rho\omega} & \cdot\omega & \cdot\omega & \cdot\omega & 0 & 0 & 0 & \frac{\rho_1}{\rho\omega} & \frac{\rho_1}{\rho\omega} \\ \cdot\omega & \cdot\omega & & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ \cdot\omega & & \cdot\omega & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ \cdot\omega & \cdot\omega & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ \cdot\omega & & \cdot\omega & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ 0 & \cdot & \cdot\omega & \cdot\omega & \frac{\rho_N}{\rho\omega} & 0 & 0 & \frac{\rho_N}{\rho} & \frac{\rho_N}{\rho\omega} \\ \frac{\rho_1 u}{\rho\omega} & \cdot\omega & \cdot\omega & \cdot\omega & \frac{\rho_N \omega u}{\rho\omega} & \tilde{l}_x & \tilde{m}_x & u+\tilde{\xi}_x \omega a & u-\tilde{\xi}_x \omega a \\ \frac{\rho_1 v}{\rho\omega} & \cdot\omega & \cdot\omega & \cdot\omega & \frac{\rho_N \omega v}{\rho\omega} & \tilde{l}_y & \tilde{m}_y & v+\tilde{\xi}_y \omega a & v-\tilde{\xi}_y \omega a \\ \frac{\rho_1 w}{\rho\omega} & \cdot\omega & \cdot\omega & \cdot\omega & \frac{\rho_N \omega w}{\rho\omega} & \tilde{l}_z & \tilde{m}_z & w+\tilde{\xi}_z \omega a & w-\tilde{\xi}_z \omega a \\ A_1 & \cdot & \cdot\omega & \cdot\omega & A_N & \tilde{\theta}_l & \tilde{\theta}_m & h_0+\tilde{\theta}_k a & h_0-\tilde{\theta}_k a \end{bmatrix}, \qquad (2.20)$$

where $A_i = \omega \frac{\rho_i}{\rho\omega}\left(\|q\|^2 - \omega \frac{1}{\tilde{\gamma}-1}\frac{\partial p}{\partial \rho_i} \right)$. The pressure derivatives are listed in the Appendix of [20]. Also, $\tilde{l}$ and $\tilde{m}$ are unit vectors orthogonal to each other and perpendicular to $\nabla\xi$.

The matrix L, whose rows are the left eigenvectors, is given by

$$R =\omega \begin{bmatrix} \frac{\rho\omega}{\rho\omega_1} - \phi\psi\omega_1 & \cdot & \cdot & \cdot\omega & -\phi\psi\omega_1 & \phi u & \phi v & \phi w \\ \cdot\omega & \cdot\omega & & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ \cdot\omega & & \cdot\omega & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ \cdot\omega & & & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega & \cdot\omega \\ -\phi\psi\omega_1 & \cdot & \cdot & \cdot\omega & \frac{\rho}{\rho\omega_N} - \phi\psi\omega_N & \phi u & \phi v & \phi w \\ -\tilde{\theta}\omega_l & \cdot & \cdot & \cdot\omega & -\tilde{\theta}\omega_l & \tilde{l}_x & \tilde{l}_y & \tilde{l}_z \\ -\tilde{\theta}\omega_m & \cdot & \cdot & \cdot\omega & -\tilde{\theta}\omega_m & \tilde{m}_x & \tilde{m}_y & \tilde{m}_z \\ -\frac{\theta_k}{2a} + \frac{\phi\psi_1}{2} & \cdot & \cdot & \cdot & -\frac{\theta_k}{2a} +\omega \frac{\phi\psi\omega_N}{2} & \frac{\tilde{\xi}_x}{2a} -\omega \frac{\phi u}{2} & \frac{\tilde{\xi}_y}{2a} -\omega \frac{\phi v}{2} & \frac{\tilde{\xi}_z}{2a} -\omega \frac{\phi w}{2} \\ \frac{\theta_k}{2a} +\omega \frac{\phi\psi\omega_1}{2} & \cdot & \cdot & \cdot\omega & \frac{\theta_k}{2a} +\omega \frac{\phi\psi\omega_N}{2} & -\frac{\tilde{\xi}_x}{2a} - \frac{\phi u}{2} & -\frac{\tilde{\xi}\omega_y}{2a} - \frac{\phi v}{2} & -\frac{\tilde{\xi}_z}{2a} -\omega \frac{\phi w}{2} \end{bmatrix}, \quad (2.21)$$

where $\phi =\omega \frac{\tilde{\gamma} - 1}{a^2}$ and $\psi_i =\omega \frac{1}{\tilde{\gamma} - 1} \frac{\partial p}{\partial \rho_i}$.

CHAPTER III

THERMOCHEMICAL AND TRANSPORT MODELS

In a high-temperature environment, the modeling of thermodynamic and chemical processes becomes a major concern. In most cases, the gas can be considered as a chemically reacting mixture of thermally perfect components. In this chapter the thermodynamic and chemical models used to describe fluid flows in chemical non-equilibrium are discussed first, followed by transport properties.

3.1 Thermodynamic Models

From the theory of thermodynamics, four types of fluid flows may be identified based on the fluid temperature and the molecular interactions within the fluid. According to Anderson[24] the four types are defined as follows:

1. Calorically Perfect Gas: by definition this gas has constant specific heats and thus a constant ratio of specific heats $\gamma = C_P/C_V$ (γ= 1.4 for air).
2. Thermally Perfect Gas: a thermally perfect gas is defined as one in which the specific heats are functions of temperature only. Consequently, γ varies with temperature only.
3. Mixture of thermally perfect gases: components are thermally perfect, and the mixture is thermally perfect if no chemistry is present. The mixture is not

thermally perfect in the presence of chemical reactions, and the enthalpy and energy of the mixture are functions of temperature and flow composition.

4. "Real" Gas: In this case, intermolecular forces must be included. A "real" gas occurs in the presence of very high pressures or low temperatures. The thermal equation of state (2.6) is no longer valid. The mixture enthalpy and energy are now functions of temperature and a second state variable (e.g. density or pressure).

A real gas is usually not considered to be chemically reacting. It should be noted that sometimes in the scientific literature the term "real gas" has been used to denote mixtures of chemically reacting, thermally perfect gases, in most cases with the added assumption of local chemical equilibrium[23]. In this study, the fluid will be considered to be a mixture of thermally perfect gases (see 3 above).

3.2 Caloric Equation of State

The thermal equation of state (2.4) relates pressure to temperature, T. The governing equations introduced in chapter 2 are expressed in terms of conserved variables, such as momentum and energy, and do not directly describe temperature, which is determined from the internal energy of the gas. The governing equations remain valid for flows where high temperatures exist, but it becomes necessary to modify the caloric equation of state from the simple, calorically perfect, low temperature model. In many cases, high temperatures cause the onset of chemical reactions, which lead to the dissociation of molecules and ionization of neutral species. For a gas consisting of a mixture of species, the caloric equation of state is computed by writing the mixture internal energy, e, introduced in equation (2.2), as the sum of the species energies, as in

$$e_{\text{int}\,ernal} =\theta \sum_{s=1}^{NS} \rho_s e_s \,. \tag{3.1}$$

For ideal gases, the relationship between internal energy and temperature is linear. However, for reacting gases, the molecules in addition to having transitional movement also have rotational and vibrational modes. At relatively large temperatures these additional effects lead to a non-linear relationship between internal energy and temperature. The general form of the equation for species energy is given by the equation

$$e_s =\theta \int_{T_{ref}}^{T} c_{v_s} (\tau\theta\, d\,\tau \,+\theta h_{f_s} \,, \tag{3.2}$$

where h_{f_s} is the species heat of formation, or the energy required to create that species at T_{ref}, and c_{v_s} is the specific heat at constant volume for species *s*. If the translational, rotational and vibrational components are assumed to be in thermodynamic equilibrium, and the vibrational mode is modeled using a simple harmonic oscillator[21], then the species energy is given as

$$e_s =\theta n_s R_s T +\theta \sum_{v=1}^{NVT_s} \frac{R_s \;{}_{v,s}}{e^{\;{}_{v,s}/T} -1} +\theta h_{f_s} \,, \tag{3.3}$$

where NVT_s is the number of the vibrational modes, $_{\text{v,s}}$ is the characteristic vibrational temperature(s), and n_s is a constant that specifies the translational and rotational contributions to internal energy. Equation (2.3), with equations (3.1) and (3.3), is the caloric equation of state.

3.3 Speed of Sound

Another thermodynamic property that is of great interest in fluid flow problems is the speed of sound, which is the speed at which disturbances are propagated throughout the flow field. For a chemically reacting flow two limiting cases arise for the calculation of the speed of sound. The first one occurs when the flow is assumed to be in local equilibrium; the second one is when the flow is assumed to be frozen, i.e. no chemical reactions are taking place. For non-equilibrium flows, the frozen speed of sound plays a dominant role in the governing equations, and is needed for the calculations. The frozen speed of sound is defined as

$$a^2 = \theta\left(\frac{\partial p}{\partial \rho\theta}\right)_{s,\rho\theta/\rho}, \tag{3.4}$$

where s is the entropy per unit mass. Using the Laws of Thermodynamics and the thermal equation of state, after some algebra, the final form for the frozen speed of sound is found to be [25]

$$a^2 = \theta\tilde{\gamma}\left(\frac{p}{\rho\theta}\right), \tag{3.5}$$

where $\tilde{\gamma}$ is the mixture ratio of frozen specific heats.

3.4 Finite rate chemistry

At this point, the inviscid governing equations are fully defined, once the species production rates, $\dot{\omega}_s$, are specified for a general chemistry model. The generic equation for a chemical reaction (assuming that NR reactions are occurring simultaneously) is given by

$$\sum_{m=1}^{NS} v'_{m,r} X_m \Leftrightarrow \sum_{m=1}^{NS} v''_{m,r} X_m \qquad \text{r =1,2,…,NR,} \tag{3.6}$$

Where X_m is a chemical species in the fluid, $v'_{m,r}$ are the stoichiometric coefficients for the reactants and $v''_{,m,r}$ are the ones for the products (both related to species *m* in reaction *r*). From stoichiometry, the rate of change for a given species can be defined as[20]

$$\dot{\omega}_i = M_i \sum_{r=1}^{NR} (v''_{i,r} - v'_{i,r}) \left[K_{f,r} \prod_{m=1}^{NS} \left(\frac{\rho_m}{M_m} \right)^{v'_{m,r}} - K_{b,r} \prod_{m=1}^{NS} \left(\frac{\rho_m}{M_m} \right)^{v''_{m,r}} \right], \tag{3.7}$$

where $K_{f,r}$ is the forward reaction rate for reaction r and $K_{b,r}$ is the backward reaction rate. The forward reaction rate is usually described, for a given reaction r, by an Arrhenius-like equation

$$K_{f,r} = C_{f,r} T^{\eta_{f,r}} e^{-\theta_{f,r}/T} . \tag{3.8}$$

The coefficients C_f, η_f and $_f$ have been obtained experimentally for many different reactions over a wide range of temperatures. Similarly, the backward rate reaction can be written as:

$$K_{b,r} = C_{b,r} T^{\eta_{b,r}} e^{-\theta_{b,r}/T} . \tag{3.9}$$

Both rates depend on accurate experimental data in order to provide valid results. The equilibrium rate constant, $K_{c,r}$ can be used in lieu of the backward rate reaction, and is related to the two previous quantities as follows

$$K_{c,r} = \frac{K_{f,r}}{K_{b,r}} . \tag{3.10}$$

Using the above equations, equation (3.7) can be rewritten as:

$$\dot{\omega}_i = \theta M_i \sum_{r=1}^{NR} (v''_{i,r} - v'_{i,r}) K_{f,r} \left[\prod_{m=1}^{NS} \left(\frac{\rho\theta_m}{M_m}\right)^{v'_{m,r}} - \theta\frac{1}{K_{c,r}} \prod_{m=1}^{NS} \left(\frac{\rho\theta_m}{M_m}\right)^{v''_{m,r}} \right]. \qquad (3.11)$$

The equilibrium rate constant, like the forward and backward rates, can be represented in an Arrhenius form using experimental data. Another option for defining the equilibrium rate, (the one used in this study,) is based on the minimization of Gibbs free energy at constant pressure and internal energy. This method is not dependent on experimental data, but on reference thermodynamic values, which are easily obtained. The method does depend on the thermodynamic model used, and will be affected by any shortcomings or assumptions included in it. The expression for the thermodynamic $K_{c,r}$ presented here are borrowed from the wok of Carey Cox[23]. The equation for $K_{c,r}$ is given as

$$K_{c,r} = \theta \left(\frac{p_{ref}}{RT}\right)^{\sum_{s=1}^{NS}(v''_{s,r}-v'_{s,r})} \exp\left[-\theta\sum_{s=1}^{NS}\Omega_s(T)(v''_{s,r}-\theta v'_{s,r})\right] \qquad (3.12)$$

In the previous equation the function $\Omega_s(T)$ is determined by the equation

$$\Omega_s(T) = \theta\frac{1}{R_sT}\left[\int_{T_{ref}}^{T} c_{p_s}(\tau)d\tau - T\int_{T_{ref}}^{T}\frac{c_{p_s}(\tau)d\tau}{\tau} + h_{f_s} - T(s_{ref})_s\right], \qquad (3.13)$$

where c_{p_s} is the specific heat at constant pressure for species *s*.

In general the final form of this equation depends on the particular thermodynamic model used. For the model stated in equation (3.3), $\Omega_s(T)$ is given as

$$\Omega_s(T) = \theta + \theta_s + \theta\sum_{v=1}^{NVT_s}\ln\left(1-e^{\frac{-\theta_{v,s}}{T}}\right) - (1+n_s)\ln(T) + \theta\left(\frac{h_{f_s}}{R_sT}\right) - \left(\frac{s_s^0}{R_s}\right), \qquad (3.14)$$

where s_s^0 is a constant of integration that is derived from measured thermodynamic properties. The value of s_s^0 for this thermodynamic model is given as

$$s_s^0 = \theta(s_{ref})_s - R_s(1 + \theta n_s)\ln(T_{ref}) - \theta\sum_{v=1}^{NVT_s}\left(\frac{v,s}{(e^{v,s/T_{ref}})T_{ref}} - \ln(1 - e^{v,s/T_{ref}})\right). \quad (3.15)$$

where s$_{ref}$ is the measured entropy at reference temperature T$_{ref}$ and pressure p$_{ref}$.

3.5 Transport Properties

The dynamic viscosity and thermal conductivity of a mixture of gases can be obtained by applying particular mixture rules, like Wilke's rule[24], to the values obtained for a single species. A popular way to obtain the species dynamic viscosity is by means of Sutherland's formula[24], which relates μ_i to the temperature:

$$\mu_i = \theta c_1 \frac{T^{3/2}}{T + \theta c_2}, \quad (3.16)$$

where c_1 and c_2 are constants for a given gas. A similar formula can be used for k$_i$. At low temperatures, or in general for a frozen flow condition, the mixture thermal conductivity can be related to the viscosity by the following expression:

$$k = \theta\frac{c_p \mu}{P_r}, \quad (3.17)$$

where the Prandtl number, P_r, is assumed constant. When the accuracy of the above procedures is not sufficient, more complex calculations of the transport coefficients can be made by assuming a model for the intermolecular forces and applying kinetic theory [21]. For polyatomic gases the thermal conductivity can be evaluated by taking into account the contributions due to the different energy modes.

The diffusion of molecules of species i in a mixture can be modeled, with good approximation, using Fick's Law, which yields

$$\rho_i \overline{V}_{Di} = \theta - \rho D_i \nabla\left(\frac{\rho_i}{\rho\theta}\right), \qquad i=1,2,.....,NS, \tag{3.18}$$

where D_i is the multi-component diffusion coefficient, related to the binary diffusion coefficients D_{ij} (for the diffusion of species i into species j) by means of the approximate expression,

$$D_i = \theta \frac{1 - \theta \frac{\rho_i}{\rho\theta} \frac{M}{M_i}}{\sum_{j=1}^{NS} \frac{\rho_j}{\rho\theta} \frac{M}{M_j} \frac{1}{D_{ij}}}, \qquad i=1,2,....NS. \tag{3.19}$$

where M is the molecular mass of the mixture.

Since the diffusion velocities obtained by equation (3.18) must satisfy the following constraint (necessary for consistency with global mass conservation):

$$\sum_{i=1}^{NS} \rho_i \overline{V}_{Di} = \theta 0, \tag{3.20}$$

the diffusion coefficients cannot be prescribed independently of each other. A simple approximate approach that satisfies equation (3.20) is the use in equation (3.18) of only one global diffusion coefficient, given by:

$$D = \theta \frac{L_e}{\rho\theta} \frac{\mu}{P_r}, \tag{3.21}$$

where the Lewis number, L_e, is assumed constant in most applications.

CHAPTER IV

LOCI: A DEDUCTIVE FRAMEWORK FOR GRAPH-BASED ALGORITHMS

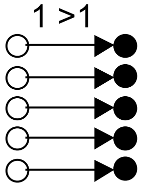
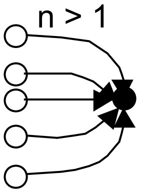
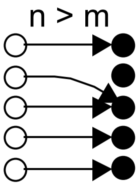
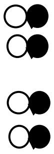
This chapter briefly describes the *Loci* framework within which the flow simulation code was developed. The material presented in this chapter is obtained from [17] and [26]. The Loci system was used to develop a chemically reacting compressible flow solver for simulating high speed and combustion flow problems (The code is called *Chem*). As part of the present effort, various flux-splitting methods were implemented in this code and tested for some standard test cases. The flux algorithms and the results will be presented in the next chapters.

4.1 The Loci Framework

The Loci system is an application framework that seeks to reduce the complexity of assembling large-scale finite-difference, finite-volume or finite-element applications, although it could be applied to many algorithms that are described with respect to a connectivity network or graph. The design of the Loci system recognizes that a significant portion of the complexity (and bugs) associated with developing large-scale computational field simulations derives from errors in control and data movement. Many of these problems are subtle, and result from gradual evolution of the application over time, giving rise to inconsistencies between various application components. The Loci framework addresses these problems by automatically generating he data and control movement operations of an application from component specifications, while guaranteeing a level of consistency between the components.

4.2 Data Models

The most fundamental concept in the Loci library is the *entity*. Entities are places where values can be stored. In Loci, these entities are given integer identifiers, and sets of entities are represented by an *entitySet*. The *entitySet* is used in Loci for control and allocation. These sets provide an efficient interface for sets of integer identifiers. In the Loci system, computational graphs are represented by collection of entities and collection of maps or connectivity lists. For example: the entities in a finite-volume calculation may represent faces, cells, and nodes of the mesh. Maps may connect faces to their left and right cells, or cells to their nodes, and so forth. Values are bound to the entities via the *store* construct, which provides an injective mapping from entities to values. The *parameter* construct provides a singleton interface to value, where a set of entities is mapped to a single value. Relationships between entities are provided by the *map* construct. The map construct can be composed with the store construct to provide an abstraction of indirection. The *constraint* construct, used to constrain computations to some subset of entities, provides an identity mapping over a given set of entities. These constructs are illustrated in figure 1.

store maps entities to values	**parameter** maps many entities to a single value	**map** maps entities to entities	**constraint** specifies a set of entities
			

These basic constructs are used to formulate a database of facts that describe the problem. The database is the fundamental starting point for logic programming systems. The definition of the problem to be solved begins as a collection of facts stored in a database, while the result of rule applications is the creation of new facts. Thus the database becomes a center of communication for programs derived from the specifications. It should be noted that, although the term database might be associated with files stored on disk, here it refers to a model of data and associated data structures. Each fact provides information about some subset of entities, such as positions of nodes, or maps relating cells to nodes. Each of these facts is given an identifier that consists of a name, an iteration label, and an iteration offset. The iteration label corresponds to the nested iteration levels of a loop.

4.3 Rule Specifications

In addition to a database of facts that includes the problem specification, a database of rules describes transformations that can be used to introduce new facts into the database. These rules correspond to fundamental computations involved in solution algorithms, such as rules for evaluating areas of faces, or for solving equations of state. These rules are specified using text strings, called rule signatures, which describe the input stores, parameters, and maps required to perform a computation, plus the list of stores or parameters that it generates. Rule signatures are of the form *head* ← *body*, where head consists of a list of variables that are generated by the application of the rule, while the body contains a list of variables that are accessed while performing the computation.

4.3.1 Rule Constraints

Any rule that is specified can be constrained to only compute values for some subset of entities. In addition to constraining rule applications, constraints also provide assertion semantics. A constraint implies that a rule must provide for every entity in the constraint. In many cases this can be used to automatically detect inconsistencies caused by incomplete information. For example, an improperly specified boundary condition yields a rule that cannot satisfy its constraint, due to insufficient information. Thus, if a boundary condition is applied to an interior node of the domain, then an error would result, caused by a conflict between the boundary condition and stencil specifications.

4.3.2 Point-wise Rules

The most common rule in finite-difference or finite-element applications is the point-wise rule. The point-wise rule represents an entity-by-entity computation of values that are placed in the stores listed in its head. The computation specified in the rule is referentially transparent and local. The semantics of the point-wise rule application requires that an output variable can only define one value per entity (this is a variation of the single assignment semantic). If an ambiguous specification produces two rules that compute values for the same entity, it is flagged as an error during scheduling. Recursion is allowed in point-wise rules, provided that the "single value per entity" criterion is not violated. Thus, recursion in point-wise rules is bounded by the number of entities in the simulation mesh.

4.3.3 Singleton Rules

A rule specification that consists exclusively of parameters for inputs and outputs is a singleton rule. Singleton rules specify a single computation that provides value for the parameters specified in its head. Like point-wise rules, these rules cannot violate the uniqueness of parameter values. Any conflicts of value assignment are reported as ambiguity errors during scheduling.

4.3.4 Reduction Rules

A reduction rule is composed of three components: a function that is applied to a set of values, an associative and commutative operator ⊕ that is defined on the type returned by the above mentioned function, and an identity element for operator ⊕, e. Thus a reduction, *r*, over values, $\{v_i \mid i \quad [1, N]\}$, using function f and operator ⊕ is defined as

$$r = \in f(v_1) \oplus f(v_2) \oplus \cdots\cdots \oplus f(v_i) \oplus \cdots\cdot \oplus f(v_N) \quad \in$$

When the reduction is evaluated using a left or right precedence rule, then a sequential evaluation is derived; however, the associative property of ⊕ allows for different parallel evaluation orders. For example, the set of values can be partitioned into subsets that can be evaluated concurrently, as in

$$r = \in \{ e \oplus f(v_1) \oplus \in f(v_2) \oplus \in \cdots\cdots \oplus f(v_p) \} \oplus \in \{ e \oplus f(v_{p+1}) \oplus f(v_{p+2}) \oplus \cdots\cdots \oplus f(v_N) \}.$$

Parallel partitioning of reduction operations can be expressed when given three basic computational methods, identified as *unit, apply,* and *join*, as listed in table 1. The *unit* rule initializes a reduction variable to the identity element, the *apply* rule "accumulates" the partial results. The algorithm for partitioning this reduction operation among parallel processors is accomplished by creating a copy of the reduction variable on each processor participating in the reduction. Each reduction variable is initialized to the identity, and then followed by the application of all *apply* rules that are in that processor's partition. Finally, the partial results for each processor are reduced to the final result using join operations.

Rule Type	Function	Rule Signature
Unit Rule	$r^0 = \mathrm{e}$	$r \leftarrow$ CONSTRAINT(v), UNIT(e)
Apply Rule	$r^{j+1} = r_i^j \oplus f(v_\mathrm{j})$	$r \leftarrow r, v,$ APPLY($\oplus$)
Join Operation	$r^{m+n} = r^m \oplus \in r^n$	Derived (No signature)

4.3.5 Iteration Rules

Iteration is defined by way of three types of rule specifications: *build* rules that construct the iteration, *advance* rules that advance the iteration, and *collapse* rules that terminate the iteration. This specification follows an analogy to the inductive proof: build rules are analogous to an inductive base, while advance rules are analogous to an inductive hypothesis. For example, an iteration where a variable named q is iterated to a converged solution may be described by the following three rules: 1) a build rule of the form $q\{n=0\} \leftarrow ic$, 2) an advance rule similar to $q\{n+1\} \leftarrow q\{n\}, dq\{n\}$, and 3) an iteration collapse rule $solution \leftarrow q\{n\}, CONDITION(converged\{n\})$. Iteration in this example proceeds by initializing the *first* iteration, *q{n=0}*, using the build rule (initial conditions). Next, termination of iteration begins with checking for the convergence: if the test succeeds then the collapse rule terminates the iteration. Finally, the iteration advances in time by the repeated application of the advance rule. Note that the completion of these rules may require invoking other rules specified in the rule database. In this case, rules that compute *converged{n}* and *dq{n}* will also need to be scheduled. To support iteration, variables that exist in lower levels of the iteration hierarchy are automatically promoted up the iteration hierarchy. Thus a variable that is computed in iteration *{n}* is communicated to iteration *{n,it}* automatically. In addition, rules that are specified completely at the stationary level will be promoted to any level of the hierarchy. This allows for the specification of relations that are iteration independent (for example, $p = \rho RT$ implies $p^n = \rho^n \in R^n T^n$).

4.4 Scheduling

In Loci, the mesh, boundary conditions, initial conditions, and other modeling information is stored in a database of facts using stores, parameters, maps and constraints, as already discussed. Scheduling occurs in three steps. The first step involves creating a dependency graph that connects the variables stored in the fact database to the goal, using the rules in the rule database. This step involves iteratively exploring the space of known variables and determining which rules can apply, which may in turn generate new variables. Once this graph is produced, it is *pruned* to only those rules that generate the requested goal, sorted into iteration hierarchies, and reduced to a directed acyclic graph (DAG) by clumping recursive dependency loops. The next step is an existential deduction phase, which determines what attributes can be assigned to which entities. For example, the rule $p \leftarrow \rho, R, T$ specifies that entities that have attributes ρ, R, and T also have the attribute *p*. The existential deduction begins with the given facts, and follows the DAG topological order, computing the entities associated with each attribute until the goal is reached. During this existential deduction, recursive loops are iteratively evaluated until all possible attributes are generated. The result of the existential deduction phase is a concurrent schedule that obtains the requested goal. However, since it is possible that some attributes may exist for entities that do not contribute to the requested goal, a final optimization step prunes this schedule. The pruning operation starts from the goal and works backwards through the DAG, until the schedule only computes those values that are needed to provide the requested goal.

The scheduling process automatically produces a concurrent schedule. Only partitioning of entities to processors is needed to generate a schedule for parallel processors (on distributed memory architectures, a communication schedule would also need to be deduced). Thus, the numerical model does not have any references to parallel execution: this derives naturally from the specification.

4.5 Implementation

The fundamental design strategy in the Loci system has been the use of shallow inheritance hierarchies, combined with templated containers and composers. As already mentioned, the most basic data type for the Loci system is the entitySet, a value class that describes arbitrary sets of entities, and provides fast intersection, union, and complement operations. These entity sets are necessary for the existential deduction phase of scheduling, and used for controlling and allocating functions. The data models described in section 4.2 are implemented as templated container classes. These containers provide features that facilitate their storage in the fact database and their automatic binding in rule invocations. The rules described in section 4.3 are implemented via a shallow inheritance hierarchy. Users create new rules by providing a constructor, which creates the rule specification, and a virtual member function called *compute*, which performs the specified computation. In Loci, a value class sequence is used to represent loop control structures. Sequences are automatically generated by the scheduling process, and may describe both sequential and parallel looping structures. The template function *do loop* provides a fast interface to the sequence control information. The combination of the two (*compute* and *do loop*), transfers the specific details of loop optimization to the Loci framework. Loci provides a register rule template that adds rules to the global rule database when constructed. This allows rules in separately compiled modules to be automatically added to the global rule database. More details regarding the rules and their functions are available in [17] and [26].

CHAPTER V

NUMERICAL FORMULATION

The governing equations presented in Chapter II are a system of hyperbolic partial differential equations. In general, there is no closed form analytical solution to these equations, which makes a numerical solution the only practical alternative. Upwind methods have become very widely used in the numerical solution of fluid dynamic equations for both perfect and chemically reacting gases. Flux-Vector and Flux-Difference Splitting are two of the most popular procedures for discretizing the inviscid fluxes in a way that best models the physics of the problem. This chapter describes the numerical formulation employed in this study. Details on the various flux-splitting methods that were implemented as part of the present efforts will be provided in Chapter VI.

5.1 Spatial Discretization

The numerical integration of equation (2.14) in Chapter 2 begins with approximations to volume and surface integrals. For the volume integrals a second order midpoint rule is used. According to this rule

$$\int_{\Omega_c} Q(x, y, z, t)dV = Q_c(t)\vartheta_c, \tag{5.1}$$

where $Q_c(t)$ is the value of Q at the cell centroid, and ϑ_c is the cell volume.

$$\vartheta_c =] \int_{\Omega_c} dV \,. \tag{5.2}$$

Cells are assumed to be fixed and rigid for simplicity. The extension to moving and deforming control volumes can be found in [17].

The numerical integration of the surface integral in equation (2.18) is accomplished by summing the contributions of each of the NF faces of cell *c*. Each individual contribution is again approximated using the midpoint rule. The inviscid flux function itself will require additional numerical treatment, and will be discussed in the next Chapter. For now, it is sufficient to assume that both inviscid and viscous fluxes can be approximated by a function, $\widetilde{F}(Q_l, Q_r)$, of conservative values to the left and right of the face. From this, F is given by

$$\int_{\partial\Omega]_c} \left(\widetilde{F} - \widetilde{F}_v\right) dS = \sum_{j=1}^{NF_c} \int_{\partial\Omega]_{c,f}} \left(\widetilde{F} - \widetilde{F}_v\right) dS \approx \sum_{j=1}^{NF_c} A_{c,f} \left[\widetilde{F}\left(Q_{l,f}, Q_{r,f}\right) - \widetilde{F}_v\left(Q_{l,f}, Q_{r,f}\right)\right), \tag{5.3}$$

where the area of the face $A_{c,f}$, is defined as

$$A_{c,f} =] \int_{\partial\Omega_c} dS \,. \tag{5.4}$$

Now equation (2.18) is numerically approximated by the equation

$$\frac{d}{dt}[\vartheta_c Q_c(t) + \sum_{j=1}^{NF_c} A_{c,f} \left[\widetilde{F}\left(Q_{l,f}, Q_{r,f}\right) - \widetilde{F}_v\left(Q_{l,f}, Q_{r,f}\right)\right) = \vartheta_c \dot{W}_c(t), \tag{5.5}$$

which is a system of ordinary differential equations of the form

$$\frac{d}{dt} Q_c = R_c \,, \tag{5.6}$$

where R_c is given by the expression

$$R_c =] \frac{1}{\vartheta]_c} \left[\vartheta_c \dot{W}_c -] \sum_{f=1}^{NF_c} A_{c,f} \left[F - F_v \left(Q_{l,f}, Q_{r,f}\right)\right) \right]. \tag{5.7}$$

Equations 5.6 and 5.7 can be used numerically to model the time evolution of the fluid when simultaneously satisfied for all cells in the mesh.

The fluxes described in the later chapter require left and right states. For first order formulations these values simply correspond to the left and the right cell values that correspond to either sides of the face. For higher order schemes, these left and right states are extrapolated from neighboring cell values. The approach that is used in the present study is applicable to meshes composed of hexahedral cells [17]. The primitive variables are extrapolated to the faces using the Monotone Upstream-Centered Schemes for Conservation Laws (MUSCL) extrapolation approach [27].

The application of high-resolution upwind schemes to solutions with discontinuities leads in general to the appearance of oscillations. Unlike central schemes, where oscillations are allowed to form and are subsequently eliminated by artificial dissipation, the approach followed for upwind schemes consists of preventing the generation of oscillations by acting on their production mechanism. In the vicinity of regions with large gradients, introducing non-linear correction factors, called limiters, modifies the extrapolation formulae. There are a variety of limiter functions that are described for conservation law equations. Barth[28] and Venkatakrishnan[29] limiters were used in the present study.

5.2 Temporal Discretization

The numerical integration of the system of equations 5.6 and 5.7 can be performed by a variety of methods. For this study, the family of schemes described by Janus[30] is used. These schemes are given by the time-discretized equation

$$\frac{(1+\psi)\Delta Q^{n}-\psi\Delta Q^{n-1}}{\Delta t}=(1-\theta)R^{n}\left(Q^{n}\right]+\theta R^{n+1}\left(Q^{n+1}\right), \qquad (5.8)$$

where

$$\Delta Q^{n}=Q^{n+1}-Q^{n}. \qquad (5.9)$$

In the above, n represents the time level and θ,ψ are two parameters that control the accuracy of the discretization. Equation (5.8) represents a set of implicit time integration schemes, including the second-order three point backward $\left(\theta=1,\psi=]\frac{1}{2}\right)$, backward Euler $(\theta=1,\psi=0)]$ and Crank-Nicholson $\left(\theta]=]\frac{1}{2},\psi=]0\right)$. It also includes explicit schemes such as forward Euler $(\theta=0,\psi=0)$.

The solution of equation (5.8) for Q^{n+1} given Q^{n} is the goal of the time integration procedure. However, the solution is complicated by the fact that that R^{n+1} (Q^{n+1}) is a non-linear function with a non-trivial inverse. Instead of solving equation (5.8) directly, a common approach is to employ the Newton iterative method for the solution of the non-linear homogeneous equation given by

$$\mathrm{Y}\left(Q^{n+1}\right)=Q^{n+1}-Q^{n}-]\frac{\Delta t}{1+\psi]}\left[(1-\theta)R\left(Q^{n}\right)+\theta R\left(Q^{n+1}\right)-]\frac{\psi}{\psi+1}\left(Q^{n}-Q^{n-1}\right)=0. \qquad (5.10)$$

The vector form of the Newton method is used to obtain the zero of the vector valued function, $\mathrm{Y}(Q)$. Consequently, the Newton method should converge to the vector value of Q^{n+1}. The Newton method proceeds by iteratively solving the equation

$$\mathrm{Y}'\left(Q^{n+1,p}\right)\left(Q^{n+1,p+1}-Q^{n+1,p}\right)=-\mathrm{Y}\left(Q^{n+1,p}\right), \qquad p\geq 0, \qquad (5.11)$$

where the Newton iteration is initialized using the previous time-step values, thus $Q^{n+1,p=0}=Q^{n}$. The Jacobian, $\mathrm{Y}'\left(Q^{n+1,p}\right)$, is given by the equation

$$\Upsilon'\left(Q^{n+1,p}\right) =] I - \frac{\theta\Delta t}{1+\psi]}\left[\frac{\partial]}{\partial Q}R\ \left(Q^{n+1,p}\right)\right]^{n+1,p} =]$$

$$I -] \frac{\theta\Delta t}{(1+\psi)}\left[\frac{\partial\dot{W}}{\partial Q} -] \sum_{f\in]faces}\frac{A_f}{\vartheta}\frac{\partial\left[\widetilde{F}\left(Q_{l,f},Q_{r,f}\right) -]\widetilde{F}_v\left(Q_{l,f},Q_{r,f}\right)\right]}{\partial Q}\right]^{n+1,p} . \qquad (5.12)$$

The Jacobian in equation 5.12 is a sparse matrix with dense sub-blocks. The off-diagonal blocks are terms generated by the flux Jacobians, due to their functional dependence on neighboring cells.

5.3 Linear System Solution

Each Newton iteration step given by equation (5.11) requires solving a linear system of equations of the form

$$Ax =]b , \qquad (5.13)$$

where,

$$A =]\Upsilon'\left(Q^{n+1,p}\right), \quad x = \left(Q^{n+1,p+1} -]Q^{n+1,p}\right), \quad b =]-\Upsilon\left(Q^{n+1,p}\right). \qquad (5.14)$$

The matrix, A, of this linear system is given by equation (5.12) and is typically a matrix with sparse structure that is composed of dense sub-blocks. Most of the terms of equation (5.12) contribute to the diagonal block of the matrix, while the flux terms, being a function of left and right Q values, contribute to off diagonal terms. Inviscid flux Jacobian terms are treated as Jacobians of the first order functions in an effort to increase the sparseness of matrix A, and similar steps are taken for the viscous flux Jacobians. Both analytical and numerical Jacobians can be employed [17]. The matrix A can be factored into a lower, upper, and diagonal blocks, as in

$$A =]L +]D + U\,. \qquad (5.15)$$

For first order flux Jacobians, each internal face of the mesh contributes a term to the diagonals of the cells to either side, while also contributing one block to the lower matrix, L, and one block to the upper matrix, U. The system of equations is solved using a symmetric Gauss-Seidel method. Iterative methods of this form are rather efficient at solving systems of equations produced by finite volume schemes applied to hyperbolic equations, such as the fluid dynamics equations presented here. The symmetric Gauss-Seidel iterative solver works by a two-pass method. These two passes, a forward and a backward pass, are a consequence of the solution of the factored equations

$$\left(L +]D\right)x^{*i+1} + Ux^{i} =]b\,,$$

and

$$Lx^{*i+1} + \left(D + U\right)x^{i+1} =]b\,, \qquad (5.16)$$

where x^{*i+1} is the result of the forward pass of the symmetric Gauss-Seidel iteration. The iteration is initialized with the first pass of a block Jacobi iterative method.

$$x^{0} =]D^{-1}b\,. \qquad (5.17)$$

In the solution of equations (5.15) and (5.16), a dense GAXPY (General A x Plus y) LU method[31] is employed to invert the diagonal blocks of D. This LU factorization is performed once before the Gauss-Seidel iteration proceeds, and is used in both passes.

CHAPTER VI

DISCRETIZATION OF INVISCID FLUXES

The numerical treatment of the inviscid flux function needs to be done carefully, in order to avoid oscillatory behavior around solution discontinuities such as shocks. Dissipation that damps high frequency signals can be added to the numerical fluxes. Unfortunately, there is a tradeoff between this damping and the accurate resolution of discontinuities. It is generally agreed that characteristic-based algorithms are better suited to resolve solution discontinuities. In the next sections several different flux-splitting techniques are discussed, which were investigated as part of this study. Only the Roe scheme is a flux-difference splitting method, while all the other schemes are flux-vector splitting methods.

6.1 Roe-Averaged Fluxes

The essential features of flux difference split algorithms involve the solution of local Riemann problems arising from the consideration of discontinuous states at cell interfaces on an initial data line. The Roe averaged flux formulation is based on a linearization of the Riemann problem, and is the most widely used approach. The form suitable for chemically reacting flow problems has been developed by Cinnella[22].

In the following, the *arithmetic* average of a quantity f will be denoted by:

$$\langle f \rangle = \frac{f_l + f_r}{2}, \tag{6.1}$$

with the subscript *l* indicating the left state and the subscript *r* the right state, respectively. Also, the jump of a quantity *f* will be defined as

$$\|f\| = f_r - f_l, \tag{6.2}$$

and the $\wedge$ notation will be used for Roe-averages, to be defined later.

The solution of the approximate Riemann problem involves the determination of the cell interfaces fluxes as a summation over wave speeds. The final result for $\hat{F}(Q_l, Q_r)$, the flux function in the -direction, for example, reads:

$$\hat{F}(Q_l, Q_r) = \langle \hat{F} \rangle - \frac{1}{2}\left(\|\hat{F}\|_A + \|\hat{F}\|_B + \|\hat{F}\|_C \right). \tag{6.3}$$

Here, the $\|\hat{F}\|_A$ term corresponds to the eigenvalue $\lambda_A = \tilde{u}$ and may be written as:

$$\|\hat{F}\|_A = J|\nabla\xi| \left(\|\rho\| - \frac{\|p\|}{\hat{a}^2} \right) |\tilde{u}| \begin{bmatrix} \hat{\rho}_1 \\ \hat{\rho}_2 \\ \vdots \\ \vdots \\ \hat{\rho}_{NS} \\ \hat{u} \\ \hat{v} \\ \hat{w} \\ \hat{h}_0 - \hat{a}^2/(\hat{\gamma}-1) \end{bmatrix} + J|\nabla\xi| \hat{\rho} |\tilde{u}| \begin{bmatrix} \|\rho_1/\rho\| \\ \|\rho_2/\rho\| \\ \vdots \\ \vdots \\ \|\rho_{NS}/\rho\| \\ \|u\| - \tilde{\ }_x \|\tilde{u}\| \\ \|v\| - \tilde{\ }_y \|\tilde{v}\| \\ \|w\| - \tilde{\ }_z \|\tilde{w}\| \\ \Theta \end{bmatrix}, \tag{6.4}$$

where

$$\Theta = \left(\hat{u}\|u\| + \hat{v}\|v\| + \hat{w}\|w\| \right) - \hat{\tilde{u}} \cdot \|\tilde{u}\| - \sum_{s=1}^{NS} \hat{\psi}_s \left\| \frac{\rho_s}{\rho} \right\| . \tag{6.5}$$

Similarly, the terms associated with $\|\hat{F}\|_B$ and $\|\hat{F}\|_C$ are given by

$$\left\|\hat{F}\right\|_{B,C} = \xi J \left|\nabla \xi\right| \frac{1}{2\hat{a}^{2}}\left(\xi\|p\| \pm \hat{\rho}\hat{a}\|\hat{u}\|\right)\tilde{u} \pm \xi\hat{a}\Big| \begin{bmatrix} \hat{\rho}_{1} \\ \hat{\rho}_{2} \\ \\ \\ \hat{\rho}_{NS} \\ \hat{u} \pm \tilde{\xi}_{x} a \\ \hat{v} \pm \tilde{\xi}_{y} a \\ \hat{w} \pm \tilde{\xi}_{z} a \\ \hat{h}_{0} \pm \xi\hat{\tilde{u}}\,\hat{a} \end{bmatrix}. \qquad (6.6)$$

Similar results hold for the flux functions in the other directions. The Roe averaged speed of sound, $\hat{a}$, is given by the equation

$$\hat{a}^{2} = \xi(\hat{\gamma}-1)\left[\hat{h}_{0} - \xi\frac{\hat{q}^{2}}{2} + \xi\hat{c}_{v}^{*}\hat{T} - \xi\sum_{s=1}^{NS}\hat{\rho}_{s}\hat{e}_{s}\right]. \qquad (6.7)$$

In the above, the Roe averaged values for the flow variables are given as

$$\hat{\rho} = \sqrt{\rho_{l}\rho_{r}}\,, \qquad \hat{u} = \xi\frac{\left\langle u\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\xi\right\rangle}, \qquad \hat{v} = \xi\frac{\left\langle v\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\right\rangle}, \qquad \hat{w} = \xi\frac{\left\langle w\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\xi\right\rangle},$$

$$\hat{h}_{0} = \xi\frac{\left\langle h_{0}\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\right\rangle}, \qquad \hat{T} = \xi\frac{\left\langle T\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\xi\right\rangle}, \qquad \hat{\rho}_{s} = \xi\frac{\left\langle(\rho_{s}/\rho)\xi\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\xi\right\rangle}, \quad \hat{R} = \xi\sum_{s=1}^{NS}\hat{\rho}_{s}R_{s}\,,$$

$$\hat{c}_{v}^{*} = \xi\sum_{s=1}^{NS}\hat{\rho}_{s}c_{v_{s}}^{*}\,, \qquad \hat{c}_{vs}^{*} = \xi\frac{1}{\|T\|}\int_{T_{1}}^{T_{2}} c_{v_{s}}\,dT\,, \qquad \hat{\gamma}-1 = \xi\frac{\hat{R}}{\hat{c}_{v}^{*}}\,, \qquad \hat{e}_{s} = \frac{\left\langle e_{s}\sqrt{\rho}\xi\right\rangle}{\left\langle\sqrt{\rho}\xi\right\rangle},$$

$$\hat{\psi}_{s} = \frac{R_{s}\hat{T}}{\hat{\gamma}-1} - \xi\hat{e}_{s} + \xi\frac{\hat{q}^{2}}{2}, \qquad \hat{\tilde{u}} = \tilde{\xi}_{x}\hat{u} + \tilde{\xi}_{y}\hat{v} + \tilde{\xi}_{z}\hat{w}. \qquad (6.8)$$

6.2 Flux-Vector Splitting

Upwind schemes compute the inviscid flux vector at a cell interface by taking into account the correct direction of propagation of information. Flux-vector splitting techniques are based upon the definition of the inviscid flux vector as the sum of two components: a positive one, which carries the information propagating from the left of the interface; and a negative one, which carries the information propagating from the right,

$$\hat{F}(Q_l, Q_r) = \xi\hat{F}^{+}(Q_l) + \xi\hat{F}^{-}(Q_r). \tag{6.9}$$

The original formulation of these techniques is based on the diagonalization and decomposition of the flux Jacobian matrix, working with the quasi-linear form of the one-dimensional Euler equations, according to the sign of the eigenvalues. In multi-dimensional flows, the flux splitting approach is based on a locally one-dimensional eigenvalue decomposition, which is equivalent to the assumption that the propagation of information is in the direction normal to the cell phases. This introduces a dependence of the solution on the mesh orientation. In the *Chem* code, a generalized grid approach was developed, which reduces the effect of the mesh on the solution[32].

6.3 Van Leer Flux Splitting

The scheme introduced by Van Leer was obtained by imposing a certain number of conditions on $\hat{F}^{+}$ and $\hat{F}^{-\xi}$ in equation 6.9. In particular, $\hat{F}^{\pm\xi}$ and the associated flux Jacobians $\partial\hat{F}^{\pm}/\partial Q$ are requested to be continuous functions of the Mach number and expressed as polynomials of the lowest possible order. In addition, the eigenvalues of $\partial\hat{F}^{+}/\partial Q$ must be positive or zero (non negative), and those of $\partial\hat{F}^{-}/\partial Q$ negative or zero (non positive). The Mach number used as a parameter in the scheme is defined as the ratio between the contravariant velocity relative to the direction under consideration and the frozen speed of sound, as follows

$$M = \xi \frac{\tilde{u}}{a} . \tag{6.10}$$

It is useful to point out that this parameter can be positive or negative, depending upon the direction of the flow relative to the interface.

When the Mach number calculated with the dependent variables associated with the left side of the interface, M_l, is larger than 1, or when the Mach number corresponding to the right side, M_r, is less than –1, a supersonic flow regime exists and the corresponding flux vector ($\hat{F}^{+}$ and $\hat{F}^{-\xi}$respectively) is simply computed using the upstream flow conditions.

In the subsonic regime, ($|M_l| \leq \xi 1$ and $|M_r| \leq \xi 1$) two splitting polynomials are introduced. The first one is used for the Mach number, and produces weighing coefficients for the mass flux, the convective portion of the momentum flux, and the energy flux. A separate splitting is used for the pressure:

$$M^{\pm\xi} = \pm\xi \frac{1}{4} \left(M_{l/r} \pm 1\right)^2 \tag{6.11}$$

$$p^{\pm\xi} = \frac{p_{l/r}}{4} \left(M_{l/r} \pm 1\right)^2 \left(2 \mp M_{l/r}\right) . \tag{6.12}$$

In the above, the + sign is for *l* and – sign is for *r*. The final expression for the inviscid flux vector at an interface is given by:

$$\hat{F}(Q_l,Q_r) = J|\nabla\xi| \left\{ M^{+} \begin{pmatrix} \rho_1 a \\ \rho_2 a \\ \cdot \\ \cdot \\ \rho_{NS} a \\ \rho u a \\ \rho v a \\ \rho w a \\ \rho h_0 a \end{pmatrix}_L + M^{-} \begin{pmatrix} \rho_1 a \\ \rho_2 a \\ \cdot \\ \cdot \\ \rho_{NS} a \\ \rho u a \\ \rho v a \\ \rho w a \\ \rho h_0 a \end{pmatrix}_R + \begin{pmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ \tilde{\xi}_x (p^{+} + p^{-}) \\ \tilde{\xi}_y (p^{+} + p^{-}) \\ \tilde{\xi}_z (p^{+} + p^{-}) \\ 0 \end{pmatrix} \right\} \tag{6.13}$$

where the metric terms correspond to the cell face under consideration. Since the original formulation of the splitting of the energy flux does not ensure the exact constancy of the total enthalpy for steady inviscid flows, an alternate treatment of the energy splitting proposed by Hanel[10] has been used in equation (6.13). The original splitting was given by:

$$\hat{F}^{\pm}_{NS+4} = M^{\pm} \left[\rho a \left(h_0 - \frac{(\tilde{u} \mp a)^2}{(\gamma + 1)} \right) \right]_{L/R}, \tag{6.14}$$

where the subscript *NS+4* indicates the last element of the split flux vector. However, the error introduced by the original formulation is of the order of the truncation error and does not seem to have any significant effect on the numerical calculations. Considering equation (6.14) to be a special case of $h_0 - m(\tilde{u} \pm a)^2$ with $m = \frac{1}{\gamma + 1}$, in the current implementation it is assumed that *m* is equal to zero.

Evaluating the derivative terms of equations (6.13) result in relatively simple expressions for the components of the split-flux Jacobians. They are presented in detail in the Appendix of [33].

6.4 Steger-Warming Flux Splitting

The governing equations introduced in Chapter 2 are hyperbolic. The Steger Warming method exploits the hyperbolic nature of the Euler equations by splitting the flux at a cell face into two parts: one part contains the information propagating upstream (associated with negative eigenvalues), and the other part contains information propagating downstream (positive eigenvalues).

The Steger Warming method uses the homogeneity of the Euler equations in order to split the fluxes (and hence the flux Jacobians matrices). Cinnella showed that homogeneity is preserved for flows with chemical non-equilibrium[22].

Extending the procedure of Cox[23], with extensions to finite-rate chemistry, the generic flux vector $\hat{F}(Q_l, Q_r)$ may be split into components (see equation (6.9)), where $\hat{F}^{\pm} = \hat{F}_A + \hat{F}_B + \hat{F}_C$, corresponding to the signs of the three distinct eigenvalues. These components are given by

$$\hat{F}_A = \xi\frac{\tilde{\gamma} - \xi 1}{\tilde{\gamma}\xi}\begin{bmatrix} \rho_1 \\ \cdot \\ \cdot\xi \\ \cdot\xi \\ \rho\xi_{NS} \\ \rho u \\ \rho v\xi \\ \rho w \\ \rho h_0 - \xi\dfrac{\rho a^2}{\tilde{\gamma} - \xi 1} \end{bmatrix}\lambda_k^{1(\pm\xi}, \xi \qquad \hat{F}_{B,C} = \frac{1}{2\tilde{\gamma}}\begin{bmatrix} \rho_1 \\ \cdot\xi \\ \cdot\xi \\ \cdot \\ \rho\xi_N \\ \rho\xi\left(u \pm a\,\tilde{}_{x}\right) \\ \rho\left(v \pm a\,\tilde{}_{y}\right) \\ \rho\left(w \pm a\,\tilde{}_{z}\right)\xi \\ \rho\xi\left(h_0 \pm \tilde{u}a\right)\xi \end{bmatrix}\lambda_k\xi^{N+3,N+4(\pm)}, \tag{6.15}$$

where now

$$\lambda_k^{1(\pm\xi} = J\left|\nabla\xi\right|\frac{\tilde{u} \pm \left|\tilde{u}\right|}{2} \quad ,$$

$$\lambda_k^{N+3(\pm)} = J|\nabla \ \ |\frac{(\tilde{u}+a)\pm|\tilde{u}+a\ |}{2} \ , \tag{6.16}$$

$$\lambda_k^{N+3(\pm)} = J|\nabla\xi|\frac{(\tilde{u}-a)\pm|\tilde{u}-a\ |}{2} .$$

The derivative terms which form the components of the split-flux Jacobians are relatively complicated, and are presented in detail in the Appendix of [34].

6.5 AUSM family

The most popular schemes that employ the flux-vector splitting approach have been the Van Leer/Hanel scheme[35] and the AUSM algorithms[11, 12]. These schemes feature several favorable properties: conservation of enthalpy for steady flows; small dissipation in the shear layer. In the present study, five different variations within the AUSM family of algorithms were implemented.

It is useful to first review briefly the basic AUSM scheme. An alternative form for the numerical flux is given in [12], as

$$\hat{F}(Q_l, Q_r) = \xi\frac{1}{2}\left[u_{1/2}(\Phi_l + \Phi_r) - |u_{1/2}|(\Phi\xi - \Phi_l)\right] + P_{1/2} , \tag{6.17}$$

where

$$\Phi = (\rho_1, \rho_2, ...\rho_{NS}, \rho u, \rho v, \rho w, \rho h_0)^{\mathrm{t}} , \tag{6.18}$$

$$P_{1/2} = \left(0,0,......0, \tilde{}_x p_{1/2}, \tilde{}_y p_{1/2}, \tilde{}_z p_{1/2}, 0\right)^t . \tag{6.19}$$

Here the interface velocity $u_{1/2}$ and pressure $p_{1/2}$ are defined as

$$u_{1/2} = u_l^+ + u_r^{-\xi}, \tag{6.20}$$

where

$$u^{\pm\xi} = \xi \begin{cases} \pm \frac{1}{4a}(\tilde{u} \pm \xi a)\xi, & if|\tilde{u}| \le \xi a \\ \frac{1}{2}(\tilde{u} \pm \xi \tilde{u}|), & otherwise \end{cases} \tag{6.21}$$

and

$$p_{1/2} = \xi p_l^{+\xi} + \xi p_r^-, \tag{6.22}$$

where

$$p^{\pm} = \xi \begin{cases} p(M \pm 1)^2(2 \mp M)/4, & if\ |\tilde{u}| \le \xi a \\ p(\tilde{u} \pm \xi \tilde{u})/2\tilde{u}, & otherwise. \end{cases} \tag{6.23}$$

6.5.1 AUSMD and AUSMV

These schemes are derived from the AUSM scheme, and “D” and “V” denote a flux-difference-splitting-biased scheme and flux-vector-splitting-biased one, respectively. More details are given in [13].

Both AUSMD and AUSMV schemes can be defined in a general form, by using mass flux $(\rho\tilde{u})$ instead of numerical flux, as follows:

$$\hat{F}(Q_l, Q_r) = \xi \frac{1}{2}\left[(\rho\tilde{u})\xi_{/2} \ (\Psi\xi + \Psi\xi) - |(\rho u)_{1/2}|(\Psi\xi - \Psi\xi)\right] + \xi P_{1/2}, \tag{6.24}$$

where

$$\Psi = (\rho\xi / \rho\xi\rho_2 / \rho, ... \rho_{NS} / \rho\xi u, v, w, h_0)^t. \tag{6.25}$$

The original AUSM is a special case of the above, because equation (6.24) reduces to AUSM when the mass flux is defined as

$$(\rho\tilde{u})\xi_2 = \frac{1}{2}\left[u_{1/2}(\rho\xi + \rho_r) - \xi u_{1/2}|(\rho\xi - \rho_l)\right]. \tag{6.26}$$

Previously Liou and Steffen proposed a different version of the AUSM scheme, which was based on Mach number splitting[11]. The present scheme also includes this different version, if the mass flux is defined as follows

$$(\rho\widetilde{u})_{1/2} = \frac{1}{2}\left[M_{1/2}\left(\rho_l a_l + \rho_r a_r\right) - \left|M_{1/2}\right|\left(\rho_r a_r - \rho_l a_l\right)\right], \tag{6.27}$$

where

$$M_{1/2} = u_l^{+}/a_l + u_r^{-}/a_r\,. \tag{6.28}$$

Equation (6.24) represents both AUSMD and AUSMV. AUSMV differs from AUSMD only in the treatment of the momentum fluxes. For AUSMV, the formula used is a modification of equation (6.24), namely,

$$(\rho u\widetilde{u})_{AUSMV} = u_l^{+}(\rho u)_l + u_r^{-}(\rho u)_r, \tag{6.29}$$

while that of AUSMD reads exactly as in equation (6.24), that is:

$$(\rho u\widetilde{u})_{AUSMD} = \frac{1}{2}\left[(\rho\widetilde{u})_{1/2}\ (u_l + u_r) - \left|(\rho\widetilde{u})_{1/2}\right|(u_r - u_l)\right], \tag{6.30}$$

and similar formulas apply to $\rho v\widetilde{u}$ and $\rho w\widetilde{u}$, respectively.

The AUSMV includes the Van Leer/ Hanel scheme[35], in which the mass flux is given as

$$(\rho\widetilde{u})_{1/2} = u_l^{+}\rho_l + u_r^{-}\rho_r\,. \tag{6.31}$$

It should be noted that for both equations (6.28) and (6.31) $u^{\pm}$ are defined in Eq.(6.21). On the other hand, the mass flux and the pressure and velocity splittings of the AUSMD and AUSMV schemes have to be introduced next.

The main drawback of the Van Leer/Hanel scheme is the numerical viscosity at contact surfaces, while that of the original AUSM is the numerical overshoot at shock waves. This overshoot is mainly due to the mass flux differencing. This is probably because the AUSM mass fluxes of equation (6.26) or equation (6.27) do not directly take into account the density behind the shock waves. All this was taken into consideration by the modified AUSMD and AUSMV schemes. (For details please refer to [13]). First the mass flux is defined as

$$(\rho\tilde{u})_{1/2} = u_l^{+}\rho_l + u_r^{-}\rho_r, \tag{6.32}$$

where the velocity splittings u_l^{+}, u_r^{-} are no longer the familiar Van Leer splittings of equation (6.21), rather include terms designed to capture stationary/moving contact discontinuities:

$$u_l^{+} = \begin{cases} \alpha_l\left\{\dfrac{(\tilde{u}_l + a_m)^2}{4a_m} - \dfrac{\tilde{u}_l + |\tilde{u}_l|}{2}\right\} + \dfrac{\tilde{u}_l + |\tilde{u}_l|}{2}, & if\ \dfrac{|\tilde{u}_l|}{a_m} \le 1; \\ \dfrac{\tilde{u}_l + |\tilde{u}_l|}{2}, & otherwise, \end{cases} \tag{6.33}$$

$$u_r^{-} = \begin{cases} \alpha_r\left\{\dfrac{-(\tilde{u}_r - a_m)^2}{4a_m} - \dfrac{\tilde{u}_r - |\tilde{u}_r|}{2}\right\} + \dfrac{\tilde{u}_r - |\tilde{u}_r|}{2}, & if\ \dfrac{|\tilde{u}_r|}{a_m} \le 1; \\ \dfrac{\tilde{u}_r - |\tilde{u}_r|}{2}, & otherwise, \end{cases} \tag{6.34}$$

where

$$\alpha_l = \frac{2(p/\rho)_l}{(p/\rho)_l + (p/\rho)_r}, \quad \alpha_r = \frac{2(p/\rho)_r}{(p/\rho)_l + (p/\rho)_r}. \tag{6.35}$$

and

$$a_m = \max(a_l, a_r). \tag{6.36}$$

Moreover, the pressure flux is

$$p_{1/2} = p_l^{+} + p_r^{-}, \tag{6.37}$$

where

$$p^{\pm}_{l/r} = \begin{cases} p_{l/r}\left(\dfrac{\tilde{u}_{l/r}}{a_m} \pm 1\right)^2 \left(2 \mp \dfrac{\tilde{u}_{l/r}}{a_m}\right)\Big/4, & \text{if } \dfrac{|\tilde{u}_{l/r}|}{a_m} \le 1; \\ p_{l/r}\dfrac{\tilde{u}_{l/r} \pm |\tilde{u}_{l/r}|}{2\tilde{u}_{l/r}}, & \textit{otherwise.} \end{cases} \tag{6.38}$$

Substitution of equation (6.32) and equation (6.37) into equation (6.24) results in the AUSMD numerical flux. In addition, if the momentum flux is replaced by equation(6.29), we have the scheme AUSMV.

6.5.2 AUSMDV: Mixture of AUSMD and AUSMV

In order to determine whether AUSMD or AUSMV was preferable, numerical experiments were conducted in [13]. From the experimental study it was seen that both the original AUSM and AUSMD schemes showed noticeable oscillations, although AUSMD was a little better than AUSM. On the other hand AUSMV gave fairly good solutions, indicating that it has a higher shock-capturing capability than AUSMD. In some other test cases, AUSMV produced serious oscillations, although it worked well when the CFL number was reduced to 0.5. Hence, a hybrid momentum flux was considered. This was called the AUSMDV scheme. In this, the momentum flux is defined as

$$(\rho u\tilde{u})_{1/2} = \left(\frac{1}{2} + s\right)(\rho u\tilde{u})_{AUSMV} + \left(\frac{1}{2} - s\right)(\rho u\tilde{u})_{AUSMD}, \tag{6.39}$$

where *s* is a switching function, which depends on the pressure gradient:

$$s = \frac{1}{2}\min\left(1, K\frac{|p_r - p_l|}{\min(p_l, p_r)}\right). \tag{6.40}$$

In this study, a constant parameter *K=10* is taken. Similar formulas for $\rho v\widetilde{u}$ and $\rho w\widetilde{u}$ can be developed.

6.5.3 AUSM$^+$: an Extension to AUSM

The AUSMDV scheme just presented improves the robustness of the AUSM family of algorithms in dealing with strong shocks. However, the "carbuncle phenomenon" appears, albeit much weaker than the one resulting from the Roe scheme, and it requires a fix. Moreover, the AUSMDV scheme does not capture a stationary shock exactly. Hence a new version called AUSM$^+$, has been derived in [15]. It has the following features: exact resolution of a stationary normal shock or contact discontinuity, positivity-preserving property, improvement in accuracy, simplicity, and easy generalization to other conservation laws.

The AUSM$^+$ algorithm can be simply summarized as follows (for details refer to [15]): given left and right states, the corresponding "Mach numbers" can defined as follows

$$\mathrm{M}_l =\xi\widetilde{u}_l/a_{1/2}\text{ , and }\mathrm{M}_r =\xi\widetilde{u}_r/a_{1/2}\text{ ,} \qquad (6.41)$$

where, the speed of sound $a_{1/2}$ will be defined shortly. Then, flux formula (6.17) can be used for AUSM+, provided the terms $u_{1/2}$ and $p_{1/2}$ are redefined, as follows:

$$m_{1/2} =\xi M^+(\mathrm{M}_l)+\xi M^-(\mathrm{M}_r),\quad\text{and}\quad u_{1/2} =\xi a_{1/2}m_{1/2} \qquad (6.42)$$

$$p_{1/2} =\xi P^+(\mathrm{M}\xi_l)p_l +\xi P^-(\mathrm{M}\xi_r)p_r\text{ ,} \qquad (6.43)$$

In the above, the split Mach numbers are defined as follows (a detailed analysis of the derivations can be found in [15]):

$$M^{\pm}(\mathrm{M}) = \begin{cases} \frac{1}{2}\left(\mathrm{M} \pm |\mathrm{M}|\right), & if\,|\mathrm{M}| \geq 1, \\ M^{\pm}_{\beta}(\mathrm{M}), & otherwise \end{cases} \tag{6.44}$$

where

$$M^{\pm}_{\beta}(\mathrm{M}) = \pm(\mathrm{M} \pm 1)^{2} \pm \beta\left(\mathrm{M}^{2} - 1\right)^{2}, \quad -\frac{1}{16} \leq \beta \leq \frac{1}{2} \tag{6.45}$$

and the split pressures are defined as :

$$P^{\pm}(\mathrm{M}) = \begin{cases} \frac{1}{2}\left(1 \pm sign(\mathrm{M})\right), & if\,|\mathrm{M}| \geq 1, \\ P^{\pm}_{\alpha}(\mathrm{M}), & otherwise \end{cases} \tag{6.46}$$

with

$$P^{\pm}_{\alpha}(\mathrm{M}) = \frac{1}{4}(\mathrm{M} \pm 1)^{2}(2 \mp \mathrm{M}) \pm \alpha\mathrm{M}\left(\mathrm{M}^{2} - 1\right)^{2}, \quad -\frac{3}{4} \leq \alpha \leq \frac{3}{16} \tag{6.47}$$

From experimental studies, it was proven that the best values of α and β for an accurate solution were equal to 3/16 and 1/8, respectively[15].

In order to achieve the unification of the velocity and Mach number splittings [11, 12], one can no longer use left and right values for the speed of sound, a_l or $a_{r,}$ instead a common one should be used. Many formulae were suggested in [15], however for reasons of simplicity the following splittings were implemented:

$$a_{1/2} = \frac{1}{2}\left(a_l + a_r\right) \,, \tag{6.48}$$

$$a_{1/2} = \sqrt{a_l a_r} \,. \tag{6.49}$$

However, numerical results are insensitive to the choice of the splittings implemented.

6.5.4 Low Speed AUSM$^+$

One more member of the AUSM family of fluxes can be derived by using the concept of “numerical speed of sound” in the construction of numerical flux[16]. This variable is shown to be responsible for the accurate resolution of discontinuities, such as contacts and shocks[36]. Moreover, this concept can be readily extended to deal with low speed and multiphase flows. For example, the numerical dissipation for low speed flows is scaled with the local fluid speed, rather than the sound speed, hence accuracy is enhanced, the correct solution is recovered, and the convergence rate improved.

It is widely known that the standard form of the compressible equations, discretized with either centered or upwind schemes, suffers from two major drawbacks as the flow speed approaches zero: (1) a drastic slowdown or level-off of convergence rate, (2) an inaccurate or even incorrect solution[36]. An effective way of dealing with the first problem is by inserting a premultiplying matrix to the time-derivative term. This is called the local preconditioning technique. Regarding the second problem, the inaccuracies in the upwind schemes are primarily due to the incorrect scaling of the dissipation terms as $M \rightarrow \xi 0$. In fact, the dissipation turns out to be scaled by the sound speed at low Mach numbers, thus yielding excessive numerical dissipation. This suggests that numerical fluxes need to be modified to correct this situation.

Preconditioning essentially alters the eigenvalues of the hyperbolic systems so that the wave speeds become more or less equalized. Using the time-derivative premultiplying matrix proposed by Weiss and Smith[37], the two-dimensional time dependent governing Euler (or Navier-Stokes) equations are cast in the following form (using Cartesian co-ordinates again for simplicity):

$$\Gamma\frac{\partial W}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0\ , \tag{6.50}$$

where W is a vector of primitive variables, $(p\ ,u\ ,v,\ T)^{\mathrm{T}}$ and all other variables have been introduced in chapter 2. The preconditioning matrix takes the form:

$$\Gamma=\begin{bmatrix} \Theta+\frac{1}{RT} & 0 & 0 & -\frac{\rho}{T} \\ u\left(\Theta+\frac{1}{RT}\right) & \rho & 0 & -\frac{\rho u}{T} \\ v\left(\Theta+\frac{1}{RT}\right) & 0 & \rho & -\frac{\rho v}{T} \\ H\left(\Theta+\frac{1}{RT}\right)-1 & \rho u & \rho v & \rho\left(C_p-\frac{h_0}{T}\right) \end{bmatrix}, \tag{6.51}$$

where

$$\Theta=\frac{1}{a^2}\left(\frac{1}{M_*^2}-1\right), \tag{6.52}$$

$$M_*^2=\min\left(1,\max\left(M^2,M_{co}^2\right)\right). \tag{6.53}$$

The cut-off parameter M_{co} is introduced to prevent a singularity at a stagnation point. It is a user-specified parameter: unfortunately, it does have some effects on the solution in some situations (the effect of M_{co} generally is minor, but could be of significance in some cases). A pressure difference term, as suggested by Weiss[38], could also be added to enhance robustness near the stagnation point. The reference quantity M_*^2 is bounded from above to unity if the local M exceeds one. As a result, the eigenvalues of the flux Jacobian of F with respect to W, i.e., $\Gamma^{-1}\,\partial F/\partial W$ are u, and

$$u\pm a=\frac{1+M_*^2}{2}\left(u\pm a\frac{\sqrt{\left(1-M_*^2\right)^2M^2+4M_*^2}}{1+M_*^2}\right), \tag{6.54}$$

where $M = \frac{u}{a}$ is the unscaled Mach number. Several remarks can be made concerning the eigenvalues of the preconditioned system. First, we have a constraint for the coefficient in equation(6.54).

$$\frac{1}{2} < \frac{1 + M_*^2}{2} \leq 1 . \qquad (6.55)$$

Moreover, the speed of sound is now re-scaled by a scaling factor $f(M;M_*)$. Thus, a new speed of sound can be defined.

$$\tilde{a} = f(M, M_*)a , \qquad (6.56)$$

$$f(M, M_*) = \frac{\sqrt{\left(1 - M_*^2\right)^2 M^2 + 4M_*^2}}{1 + M_*^2} . \qquad (6.57)$$

The scaling factor is also bounded,

$$1 \geq f \geq \begin{cases} |M|, & if \quad 1 >> M^2 >> M_{co}^2, \\ 2M_{co}, & if \quad 1 >> M_{co}^2 >> M^2, \end{cases} \qquad (6.58)$$

which is, the scaling factor f is bounded from below by the smaller of the local and cut-off Mach numbers.

Now equipped with a newly defined numerical speed of sound, we can readily incorporate it into the formulation of the AUSM-family schemes. The mass flux of the AUSM$^+$ scheme now can be rewritten by using the numerical speed of sound, $\tilde{a}_{1/2}$ defined by equation (6.56), in place of $a_{1/2}$ of either equations (6.48) or (6.49). The rest of the procedure is same as that of the AUSM$^+$ scheme. More details can be obtained from [16].

CHAPTER VII

RESULTS AND DISCUSSION

To validate the code, five different test cases were considered. The intent is to compare the accuracy, efficiency, and robustness of the various flux-splitting schemes described in the previous chapter. The test cases encompass a wide variety of flow conditions, ranging from the one-dimensional unsteady flow in a shock tube to the axi-symmetric steady viscous hypersonic flow past a cone. Other cases considered include the inviscid steady flow over a blunt cone, and the impingement of a convergent-divergent nozzle jet onto a flat plate. Viscous calculations were also made for a NACA airfoil at very low speed, transonic, and supersonic regimes.

7.1 Shock Tube

The simulation of a shock tube is a relatively simple test case, which involves several physical phenomena such as shocks, expansions, and contact surfaces. Consequently, it is suitable for evaluating simulations of the unsteady propagation of discontinuities. The exact solution to this particular problem, for an ideal gas model, can be obtained from the system of one-dimensional Euler equations. Experimentally, a shock tube flow can be realized by the sudden rupture of a diaphragm in a long tube separating two regions (driver and driven), filled with gases at different initial pressures (and possibly density). After removing the diaphragm, the pressure discontinuity propagates in the low-pressure region, and simultaneously an expansion fan propagates in

the high-pressure gas. In addition, a contact discontinuity separating the two gas regions propagates towards the low-pressure region of the tube.

The computational grid for the shock tube simulations employed 1000 cells equally distributed between the driver and the driven regions at *t=0*. The initial conditions on the driver side (left) and driven side (right) of the shock tube are:

Driver Side: $1*10^6$ Pa;	T = 3000.0 K;	u = 0.0 m/s
Driven Side: $1*10^5$ Pa;	T = 3000.0 K;	u = 0.0 m/s

Both regions are assumed to contain the same gas. To maintain time accuracy, a second order three point backward time integration was performed using three Newton iterations. In addition, four Gauss-Seidel iterations were performed in the iterative solver. The Barth limiter[28] was used. Both the ideal gas and a dissociating oxygen model[21] were used to simulate the shock tube scenario. All the results from the different flux splitting methods are compared with the Roe scheme, which reproduces the exact solution with good accuracy[1].

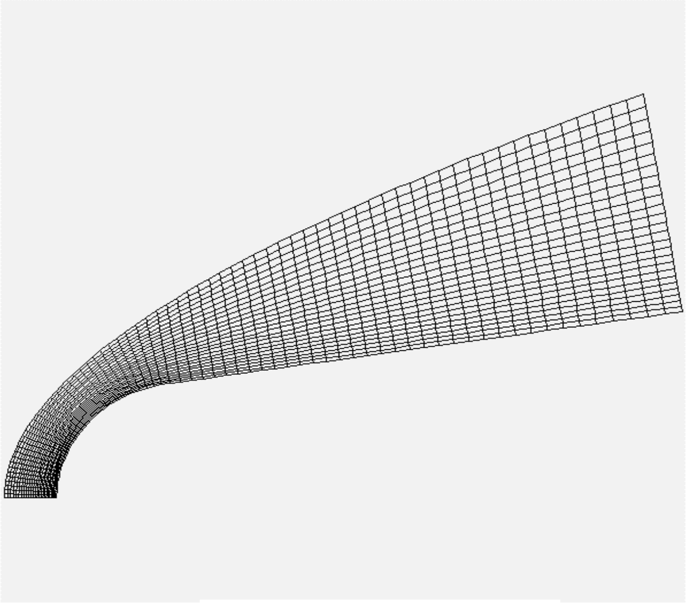
The shock tube calculations were continued until the shock reached the solid wall at one end and reflected in the opposite direction. A solid (impermeable) wall boundary condition was imposed at the end of the tube. Figures 1 to 64 show the density, Mach number, pressure and temperature plots for all the different schemes, before and after the shock has reached the solid wall, for both ideal and dissociating oxygen models. In most cases, all the schemes featured satisfactory results. However in some cases the contact surface was captured with some oscillations.

Figures 1 to 8 compare the Steger-Warming and Van Leer methods with the Roe scheme. It can be seen that the results were almost identical. Figures 9 to 16 compare the AUSMD and AUSMV schemes to the Roe scheme. It can be seen that some oscillations are present in the case when the shock reaches the wall and is reflected back, especially the AUSMD scheme. Figures 17 to 24 compare the AUSMDV scheme to the Roe scheme. This scheme has more oscillations, when the shock reached the wall and reflected back. Figures 25 to 32 feature the AUSM$^+$ and Low Speed AUSM$^+$ schemes. These algorithms have fewer oscillations when compared with other AUSM schemes. Figures 33 to 64 are similar plots in which the dissociating oxygen chemistry model was used. The relative behavior of all the schemes was roughly unchanged by the presence of chemical reactions.

Finally, the test case was run for the low speed AUSM$^+$ scheme and changing the value of the parameter M_{co}. Figures 65 to 68 compare the scheme for two different values of M_{co} (0.1 and 0.01 respectively). It can be seen that when the value is 0.1 the oscillations in the plots are reduced.

7.2 Mach 10 Blunt Cone

The second case that is used to evaluate the flux-splitting schemes is a hypersonic blunt-body case. A blunted 9-deg half-angle cone defines the geometry. The nose radius for this geometry is $r = 6.35\ cm$. The free stream conditions are given by a Mach 10 flow with an ambient pressure of $p=2.65*10^4\ Pa$ and temperature of $T=223\ K$. A second order three point backward time integration was performed using one Newton iteration. In addition, four Gauss-Seidel iterations were performed. The Barth limiter[28] was used. The grid used has 71 points distributed along the solid boundary and 26 points in the normal direction (Figure I). Two cases were run: 1) an ideal gas model, and 2) air with 5 species and 17 reactions [39].



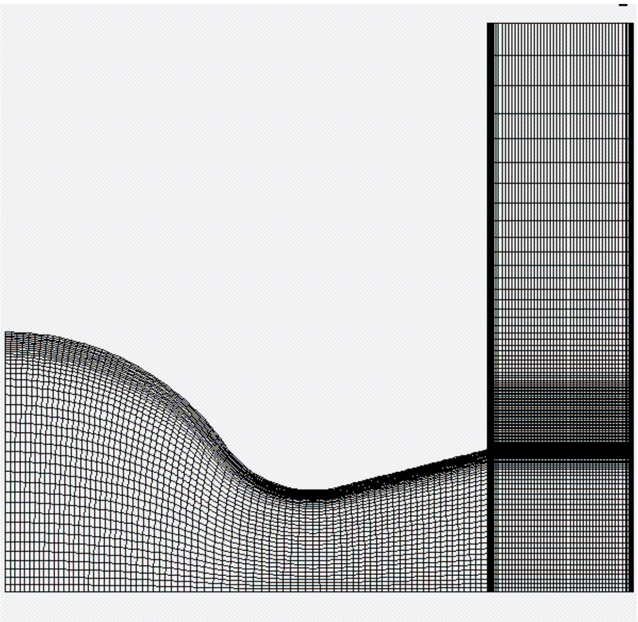
Mach 10 Blunt Cone
Figure I

Figures 69 to 74 show temperature profiles along the stagnation streamline, and figures 75 to 80 depict the variation along the body surface, respectively. Both temperature levels and shock locations are strongly affected by the inclusion of real-gas effects. It can be seen from figures 70 and 73 that the Van Leer and AUSM$^+$ schemes behave almost exactly like the Roe scheme. Figures 69 and 75 show that the Steger-Warming method is highly dissipative and deviated more from the Roe scheme. AUSMV and AUSMDV schemes deviated slightly from the Roe scheme, while the Low Speed AUSM$^+$ scheme deviated more in the case when the ideal gas model was used.

For completeness, figures 81 to 108 show the Mach number and temperature contours for all the schemes, using both the ideal gas and the chemically reacting air models.

7.3 Impingement of a Convergent–Divergent Nozzle Exhaust

Simulations were carried for a convergent-divergent nozzle in order to study the jet impingement on a perpendicular plate (2-D), and an inclined plate (3-D)[40].



Convergent-Divergent Nozzle
Figure II

7.3.1 Jet Impingement on a perpendicular plate

The convergent-divergent nozzle is shown in figure II. A three-block structured grid is used for this case. The grid is clustered near the impingement plate. The distance between nozzle and perpendicular plate was set to 0.5D, where D is the nozzle exit diameter. The air jet is channeled from a high-pressure reservoir into the atmosphere and impinges on the flat plate. The ratio between reservoir pressure (p_0) and ambient pressure (p_a) for this case is 20. An Euler time integration scheme was used, with one Newton iteration per time step. In addition, four Gauss-Seidel iterations were performed. The Venkatakrishnan limiter[29] was used.

The pressure distribution along the wall distance is presented in Figures 109, 110 and 111. Figures 112 to 118 compare all the schemes with the experimental results. In these plots the wall pressure is plotted against the non-dimensionalized distance (ratio of the distance along the wall to the radius of the jet). The density and temperature contours for all the different schemes are presented in figures 119 to 132. From these figures it can be seen that there are discrepancies in shock structure and location between predictions from Steger-Warming, AUSM$^+$ and Low Speed AUSM$^+$ on one hand, and Roe, Van Leer, AUSMV and AUSMDV on the other. The experimental results are not truly conclusive as to which schemes do a better job for this case. AUSM$^+$ and Low Speed AUSM$^+$ predicted the wrong location of the shock even when the iterations were started from a converged solution (Van Leer's method).

7.3.2 Jet Impingement on an inclined plate

The distance between nozzle exit and the center of the plate is chosen to be $2D_N$, where D_N is the nozzle exit diameter. The angle that the plate makes with the nozzle axis is taken to be 30^o.The pressure ratio between nozzle exit pressure (p_e) and ambient pressure (p_a) is chosen to be 1.2.

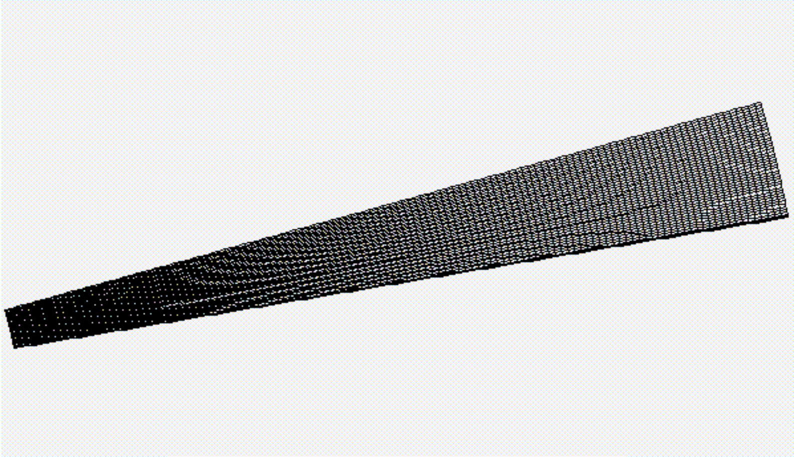
Figures 133 to 139 show the distribution of the plate wall pressure for the different schemes, compared with the experimental results (the latter are obtained from Lamont & Hunt[41]). From these it can be seen that the behavior of Roe and Van Leer schemes was similar to that of the experimental results. Also, the results from the AUSMV, AUSMDV, $AUSM^+$ and Low Speed $AUSM^+$ schemes are in reasonable agreement with the experimental results, while the Steger-Warming method deviates the most.

7.4 Hypersonic Conical Flow

This case involves the study of viscous hypersonic flow around a cone. The code was run using both a fine and a coarse grid, as shown in figures III and IV, respectively. The freestream conditions used for the calculations correspond to hypersonic flow, with M_∞ = 7.95 and a Reynolds number of 420,000, (based on the distance from the apex). The circular cone had a 10-degree half-angle. A Prandtl number equal to 1 was chosen, so that the theoretical adiabatic wall temperature

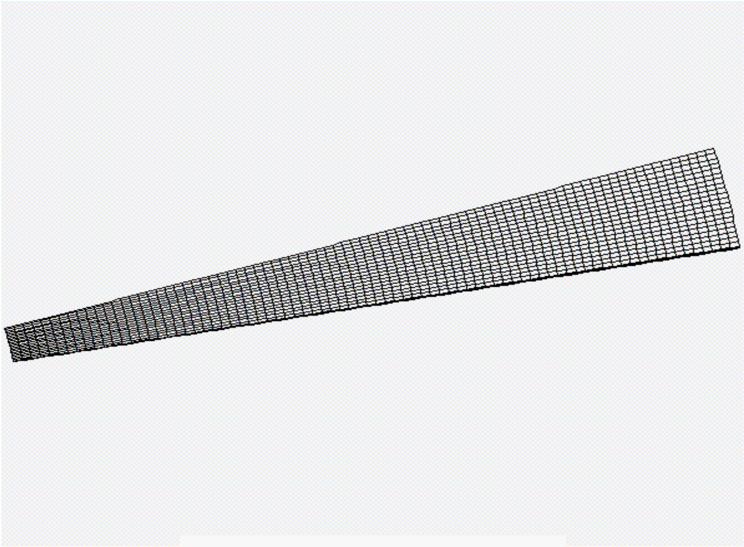
$$\frac{T_w}{T_{\infty\pi}} =\pi 1 +\pi \frac{\gamma-1}{2} M_\infty =\pi 13.64$$

could be used to check the accuracy of the numerical solutions. The grids used for calculations are composed of 41 points in the streamwise direction and 81 in the normal direction (coarse grid) and 61 points in the streamwise direction and 121 in the normal direction (fine grid). The Euler time integration was selected, using one Newton iteration per time step. In addition, five Gauss-Seidel sub-iterations were performed. The Barth limiter[28] was used. A Sutherland model was used for transport properties.



Hypersonic Conical Flow
Figure III

Figures 140 to 145 show pressure and temperature distributions as a function of angle θ, at a distance from the apex of the cone corresponding to the given Reynolds number, roughly at the mid-point of the grid. Figures 146 to 151 show the same variables for the coarse grid simulation. It can be seen that when the grid is refined, all the schemes tend to behave similarly to the Roe scheme. However, when the grid was coarser, the AUSMV and AUSMDV schemes deviated more from the Roe scheme. In both cases, Steger Warming deviated the most because it is highly dissipative.

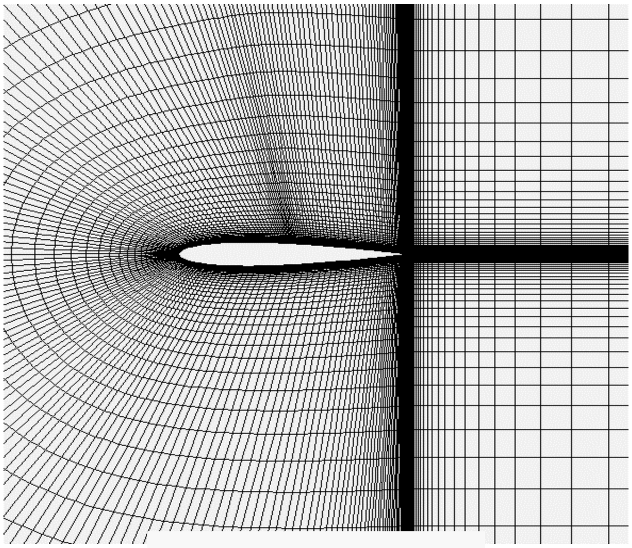


Hypersonic Conical Flow
Figure IV

7.5 NACA 0012 Airfoil

The final case used to evaluate the flux splitting schemes was the flow around a NACA 0012 airfoil. A portion of the grid close to the airfoil is shown in figure V. The grid employed for the present calculations is a C-type grid, composed of 290x81 points. Four separate flow conditions were considered: a) M_∞ = 0.799 and angle of attack = 2.86 degrees (transonic flow); b) M_∞ = 1.2 and angle of attack = 0.0 degrees; c) M_∞ = 0.1 and angle of attack = 2.86 degrees; and d) $M_{\infty\pi}$= 0.01 and angle of attack = 2.86 degrees. Cases c) and d) were considered in order to study the behavior of the low speed AUSM$^+$ scheme at such low speeds.

Figures 152 to 158 show the pressure contours for the various schemes for case a). Next, a relative error $\left(error = \pi \frac{\left| p_{other} - \pi p_{roe} \right|}{\left| p_{roe} \right|} \right)$ was calculated with respect to the Roe scheme and its contours were generated and shown in figures 159 to 164. From these figures it can be seen that the shock location was different when different schemes were implemented. Thus in the error contours large errors can be seen at two different locations, one corresponding to the Roe scheme shock location, and the other corresponding to the location predicted by the other scheme. Also some error can be seen in the lower surface of the airfoil.



NACA0012 Airfoil
Figure V

Figures 165 to 171 show the pressure contours for the supersonic case (case b). All the schemes performed quite similarly, capturing both the bow shock and the fish-tail shock. Figures 172 and 173 show the pressure contours when the Mach number is equal to 0.1 (case c). The pressure coefficient was calculated, and its contours are also shown in figures 174 and 175. It can be seen that the low speed AUSM$^+$ scheme works fine when the Mach number is small. However, when the Mach number is further reduced to 0.01 both the Roe and the Low Speed AUSM$^+$ scheme behave abnormally. This can be seen from figures 176 and 177.

Additionally, lift coefficients were calculated for cases a, c, and d. The values are tabulated as follows:

Mach Number	Roe Scheme	Low Speed AUSM^{+}	Theoretical
0.799	0.3033	0.3601	0.38
0.1	0.252	0.329	0.32
0.01	0.21	0.3505	0.318

From [42] it can be estimated that the lift coefficient of an airfoil is equal to $\frac{2\ \alpha}{\sqrt{1-\pi M_{\infty\pi}^{2}}}$, where $M_{\infty\pi}$is the free stream Mach number and α is the angle of attack. For the above experiments the angle of attack was 2.86 degrees. Therefore, the low-speed lift coefficient according to the above formula is roughly 0.32. It can be seen from the table that at low speeds (M=0.1 in particular) the low speed AUSM^{+} scheme tends to behave more accurately than the Roe scheme. However, using the Low Speed AUSM^{+} is not as effective as employing a preconditioning method[43].

CHAPTER VIII

SUMMARY AND CONCLUSIONS

The present work is concerned with enhancing the accuracy and robustness of CHEM[17], a three-dimensional flow solver which is able to simulate a wide range of flow conditions, including chemical non-equilibrium. The governing equations considered in the present study are an extension of the Navier-Stokes equations for a perfect gas. A finite volume approach has been used for the spatial discretization of the integral form of the conservation equations. The three-dimensional flow solver can be easily applied to the special cases of axisymmetric, two-dimensional, and one-dimensional flows.

Seven different flux-splitting techniques have been implemented for the discretization of the inviscid fluxes, and high-order accuracy has been achieved by means of the MUSCL extrapolation applied to primitive variables. Moreover, analytical Jacobians for the Steger-Warming and Van Leer methods were implemented. These reduce the cost of computations, as well as improving the efficiency of the flow solver.

A systematic comparison of all the different schemes with the Roe scheme is the main feature of the present work. The overall goal is to find a numerical scheme that can meet some stringent specifications of efficiency, accuracy and robustness on the widest possible spectrum of flow conditions.

The Roe scheme is very accurate, especially in the presence of discontinuities, but is not robust enough, as seen in the case of a very-low speed airfoil (when M=0.01). The Van Leer scheme seems to be more robust, but at the cost of reduced accuracy, especially for viscous problems, where an excessive numerical dissipation may significantly degrade the solution. The Steger Warming algorithm is highly dissipative, and that can be seen especially from the blunt-body cases.

The AUSMD scheme presented spurious oscillations for the shock tube test case. It is not very robust as the residuals didn't converge in other cases, and as a result no solution was obtained. AUSMV and AUSMDV behaved similarly in most cases. However, in some cases AUSMDV produced a very accurate solution, while AUSMV reported some oscillations. AUSM$^+$ and the Low Speed AUSM$^+$ behaved very similarly to the Roe scheme. However, for the case of a jet impingement on a perpendicular plate, the two schemes couldn't locate the position of the shock accurately. At low Mach numbers the Low Speed AUSM$^+$ scheme tends to behave more accurately than the Roe scheme. This can be seen in the case of an airfoil when M=0.01. However, using the Low Speed AUSM$^+$ is not as effective as employing a preconditioning method[43].

Further work can be done to eliminate the oscillatory behavior found in some of the solutions obtained with the AUSM family of schemes, possibly by using different splitting polynomials for pressure and velocity. Also, a pseudo time can be introduced as an iteration strategy, so that time accuracy for solving the unsteady equations can be better preserved.

Also the code was run with higher CFL numbers for better performance. Roe, Steger-Warming and Van Leer methods were robust enough to run at very high CFL numbers (upto 10000), while the AUSM schemes could run for a maximum CFL number of 19. Also the time performance of the code can be increased when analytical jacobians were used.

Additional work to enhance the performance of the code includes incorporating analytical Jacobians for the AUSM family, which would be less computationally expensive than the presently used numerical Jacobians. Finally, due to the overwhelming number of calculations involved in a non-equilibrium simulation, the performance of the code must be improved by considering the scheduling costs and optimizing its performance on parallel computer architectures.

REFERENCES

1. Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, vol.43, 1981, pp.357-372.

2. Chakravarthy, S. R. and Osher, S., “A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws,” AIAA Paper 85-0363, January 1985.

3. Quirk, J. J., “A Contribution to the Great Riemann Solver Debate,” ICASE Report 92-64, 1992.

4. Harten, A., Lax, P. D. and Van Leer, B., “On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws,” *SIAM Review* vol.25, 1983, pp.35-61.

5. Einfieldt, B., “On Godunov-Type Methods for Gas Dynamics,” *Journal of Numerical Analysis,* vol.25, 1988, pp.357-393.

6. Einfieldt, B., Munz, C. C., Roe, P. L. and Sjogreen, B., “On Godunov-Type Methods near Low Densities,” *Journal of Computational Physics,* vol.92, 1991, pp.273-295.

7. Wada, Y., “On the Godunov-Type Schemes-An Improvement of HLLEM Scheme and its Extension to Chemically Reacting Flows,” NAL TR-1189, 1993(in Japanese).

8. Steger, J. L. and Warming, R. F., “Flux-Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods,” *Journal of Computational Physics,* vol.40, 1981, pp.263-293.

9. Van Leer, B., “Flux-Vector Splitting for the Euler Equations,” *Lecture Notes in Physics.*, vol.170, 1982, pp.507-512.

10. Hanel, D., Schwane, R., and Seider, G., “On the Accuracy of Upwind Schemes for the Solution of the Navier-Stokes Equations,” AIAA Paper 87-1105CP, 1987.

11. Liou, M.-S. and Steffen, C.J., “A New Flux Splitting Scheme,” NASA TM104404, 1991.

12. Liou, M.-S., “On a New Class of Flux Splittings,” *Lecture Notes in Physics.*, vol.414, 1993, pp.115-119.

13. Wada, Y., and Liou, M.-S., “A Flux Splitting Scheme with High-Resolution and Robustness for Discontinuities,” AIAA Paper 94-0053, Jan.1994.

14. Liou, M.-S., “Probing Numerical Fluxes: Mass Flux, Positivity, and Entropy-Satisfying Property,” AIAA Paper 97-2035, June.1997.

15. Liou, M.-S., “A Sequel to AUSM: AUSM^{+},” *Journal of Computational Physics.*, vol.129, 1996, pp.364-382.

16. Liou, M.-S. and Edwards, J.R., “Numerical Speed of Sound and its Application to Schemes for all Speeds,” AIAA Paper 99-3268, 1999.

17. Luke, Edward, “ A Rule-Based Specification System for Computational Fluid Dynamics,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1999.

18. Cox, C. F., Cinnella, P., “General Solution Procedure for Flows in Local Chemical Equilibrium,” AIAA Journal, vol.32, No. 3, March 1994.

19. Thompson, J.F., Warsi, Z. U. A. and Mastin, C. W., *Numerical Grid Generation*, North Holland, New York, 1985.

20. Busby, Mark, “Steps Toward More Accurate and Efficient Simulations of Reactive Flows,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1997.

21. Vincenti, W. G., and Kruger, C. H., *Introduction to Physical Gas Dynamics*, John Wiley and Sons, Inc. New York, 1965.

22. Cinnella, Pasquale, “Flux-Split Algorithms for Flows with Non-Equilibrium Chemistry and Thermodymanics,” Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1989.

23. Cox, C., “An Efficient Solver for Flows in Local Chemical Equilibrium,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1992.

24. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Co., McGraw-Hill Book Co., New York, 1984.

25. Grossman, B. and Cinnella, P., “Flux-Split Algorithms for Flows with Non-Equilibrium Chemistry and Vibrational Relaxation,” *Journal of Computational Physics.*, vol.88, No.1,1990, pp.131-168.

26. Luke, Edward, “Loci: A Deductive Framework for Graph-Based Algorithms,” ISCOPE'99, Copyright Springer-Verlag, pp.142-153.

27. Van Leer, B., “Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov’s Method,” *Journal of Computational Physics*., vol.32,1979, pp.101-136.

28. Barth, T. J., and Jespersen, D. C., “The Design and Application of Upwind Schemes on Unstructured Meshes,” AIAA Paper 89-0366, 1989.

29. Venkatakrishnan, V., “On the Accuracy of Limiters and Convergence to Steady State Solutions,” AIAA Paper 93-0880, 1993.

30. Janus, J. “Advanced 3-D Algorithm for Turbomachinery,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1989.

31. Golub, G. H. and Loan, C. F. V., *Matrix Computations*, Second Edition, Johns Hopkins University Press, 1989.

32. Koomullil, R. P., “Flow Simulation System for Generalized Static and Dynamics Grids,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1997.

33. Bergamini, Lorenzo, “Towards More Robust and Efficient Upwind Solvers for all Mach Numbers: A Preliminary Study,” Master’s Thesis, Mississippi State University, Mississippi State, MS, 1993.

34. Westmoreland, William, “Numerical Solution of the Euler Equations with Non-Equilibrium Chemistry,” Master’s Thesis, Mississippi State University, Mississippi State, MS, 1997.

35. Van Leer, B., “Flux-Splitting for the 1990s,” NASA CP-3078, 1991, pp.203-214.

36. Van Leer, B., Lee, W. T., Roe, P. L., “Characteristic Time Stepping or Local Preconditioning of the Euler Equations,” AIAA Paper 91-1552, 1991.

37. Weiss, J. M., and Smith, W. A., “Preconditioning Applied to Variable and Constant Density Time-Accurate Flows on Unstructured Meshes,” AIAA Paper 94-2209, 1994.

38. Weiss, J. M., Maruszewski, J. P., and Smith, W. A., “Implicit Solution of the Preconditioned Navier-Stokes Equations using Algebraic Multigrid,” AIAA Journal 37, 29 (1999).

39. Kang, S. W., Dunn, M. G., “Theoritical and Measured Electron-Density Distributions for the Ram Vehicle at High Altitude,” AIAA Paper 72-689, 1972.

40. Cinnella, P., Luke, E., Tong, X., Tang, L., Wu, J., “Numerical Simulations of jet Impingement and Jet Separation,” MSSU-ASE-00-2, Misissippi State University, 2000.

41. Lamont, P.J. and Hunt, B. L., “The Impingement of Underexpanded, Axisymmetric Jets on Perpendicular and Inclined Flat Plates,” *Journal of Fluid Dynamics*, 100:471-521, 1980.

42. Zaccanti, M., “Analysis and Design of Preconditioning Methods for Euler Equations,” Ph.D. Dissertation, Mississippi State University, Mississippi State, MS, 1999.

43. Zaccanti, M. R., Cinnella, P., “Effective Preconditioning Methods for the Euler Equations,” AIAA Paper 00-2253, June 2000.

APPENDIX

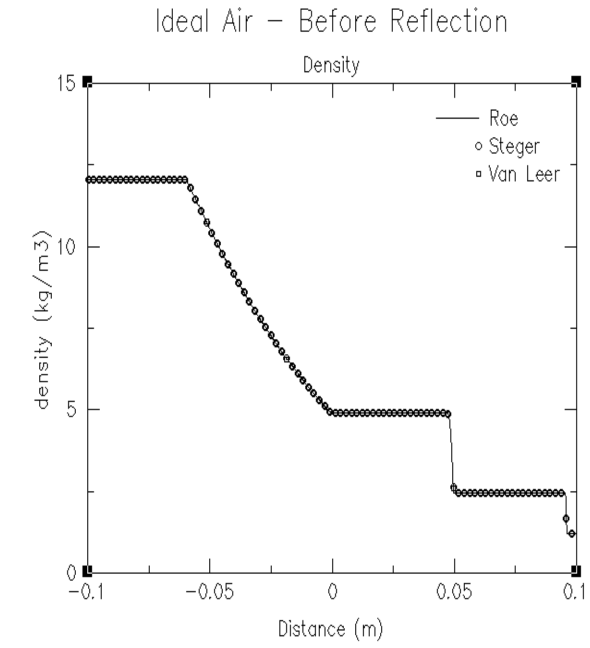


Figure 1

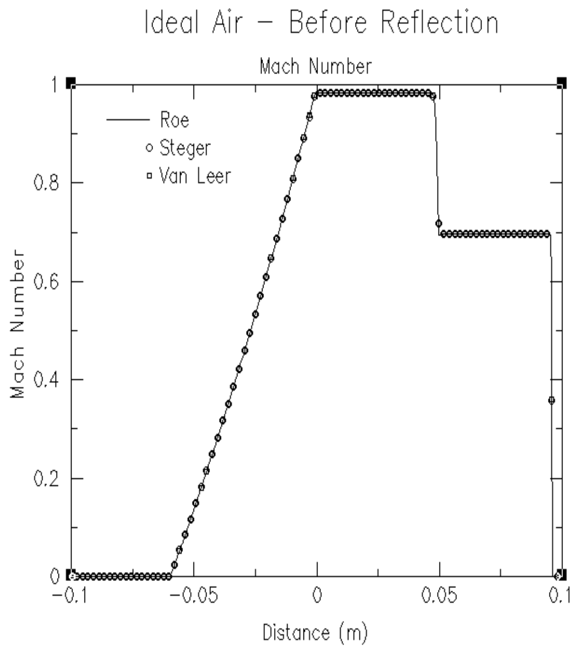


Figure 2

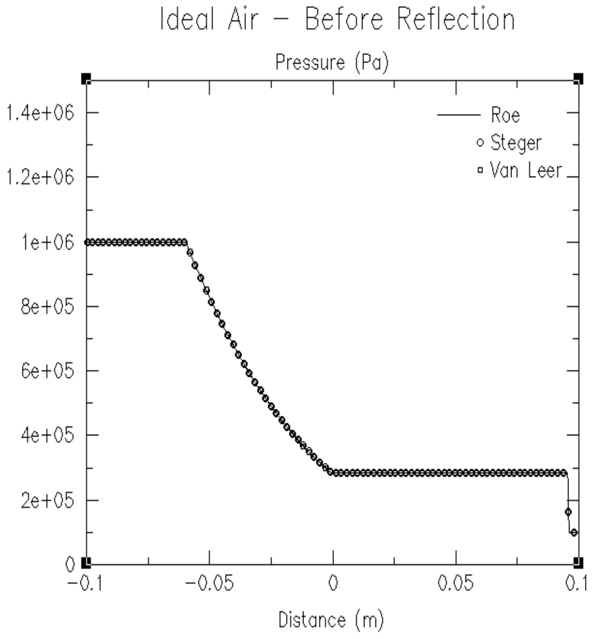


Figure 3

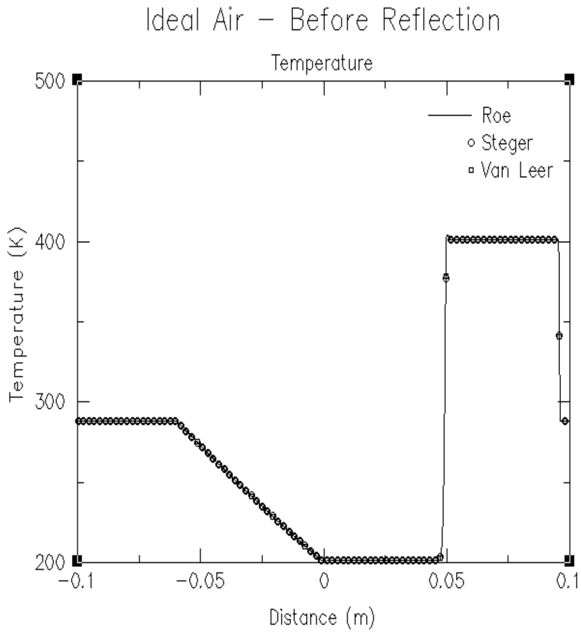


Figure 4

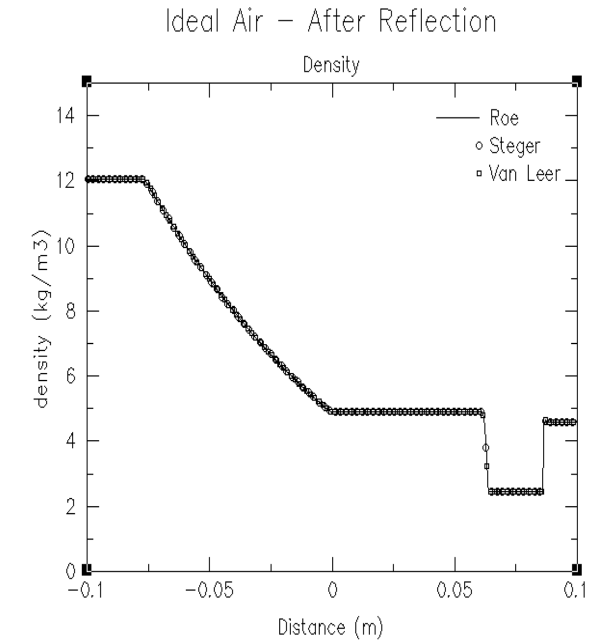


Figure 5

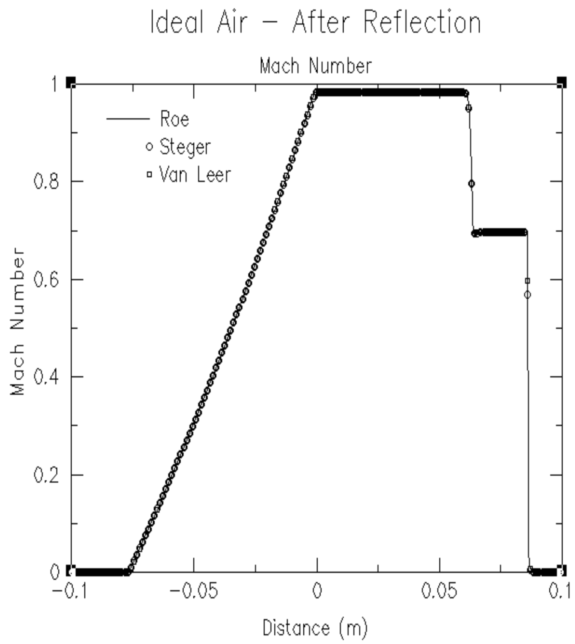


Figure 6

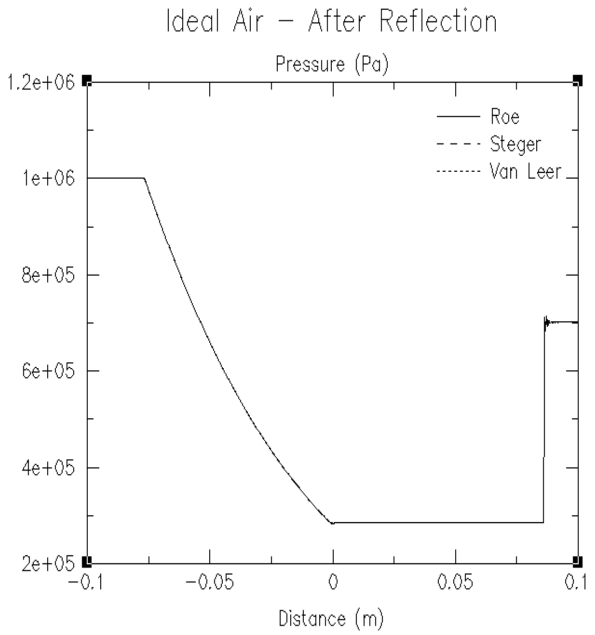


Figure 7

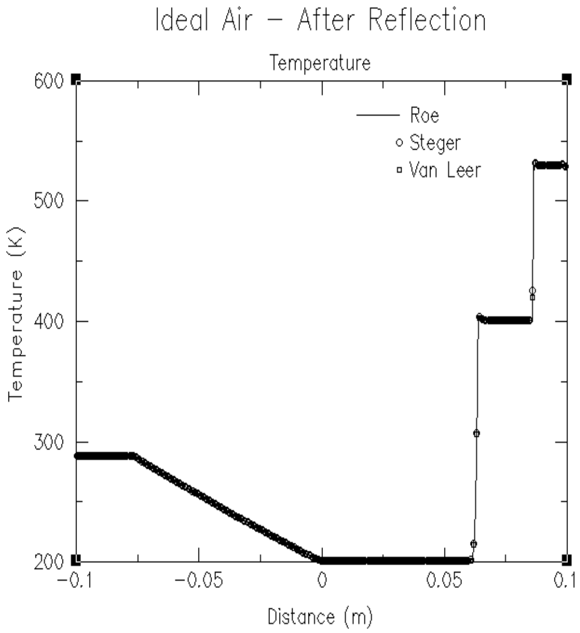


Figure 8

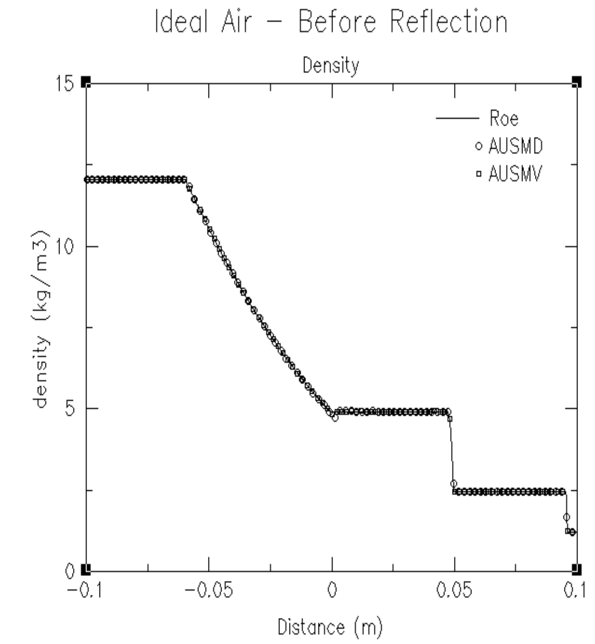


Figure 9

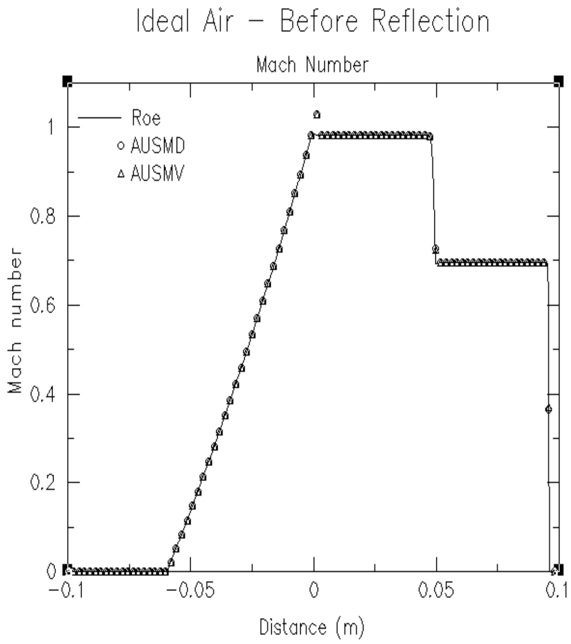


Figure 10

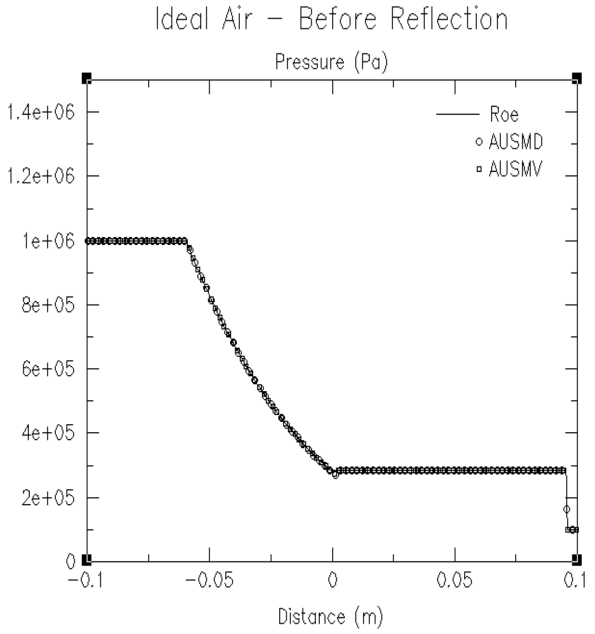


Figure 11

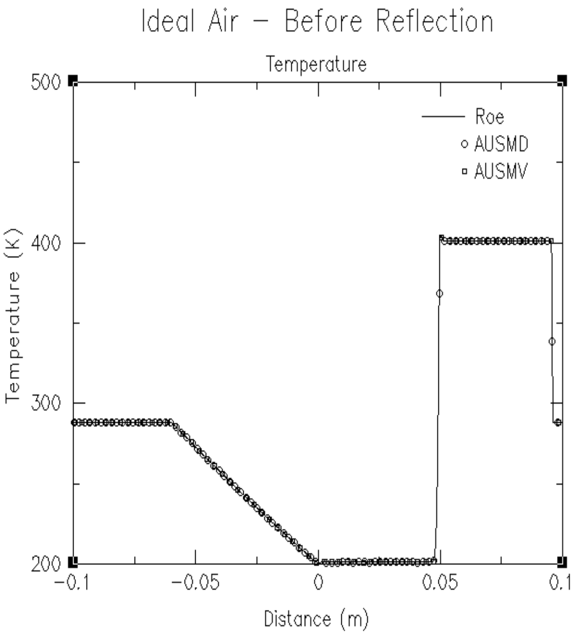


Figure 12

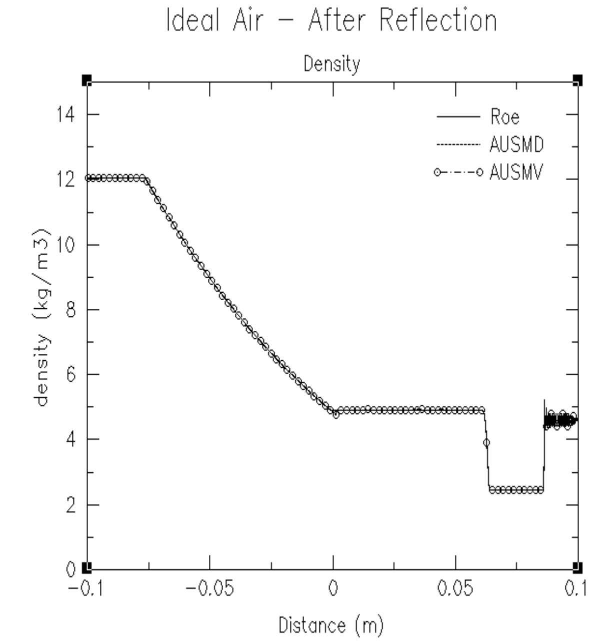


Figure 13

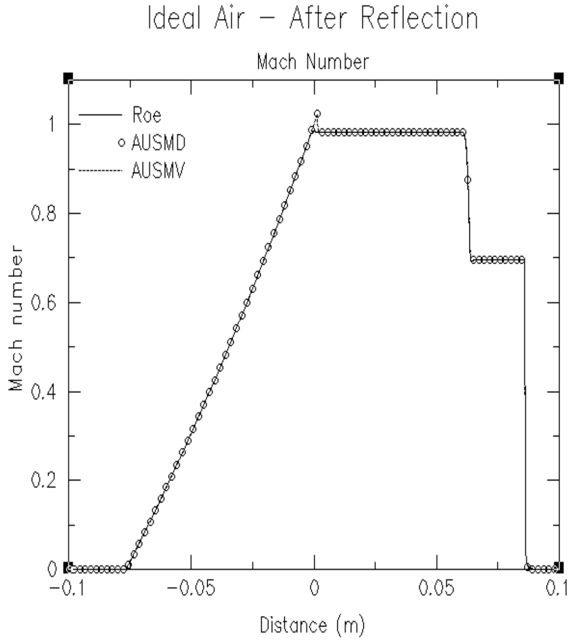


Figure 14

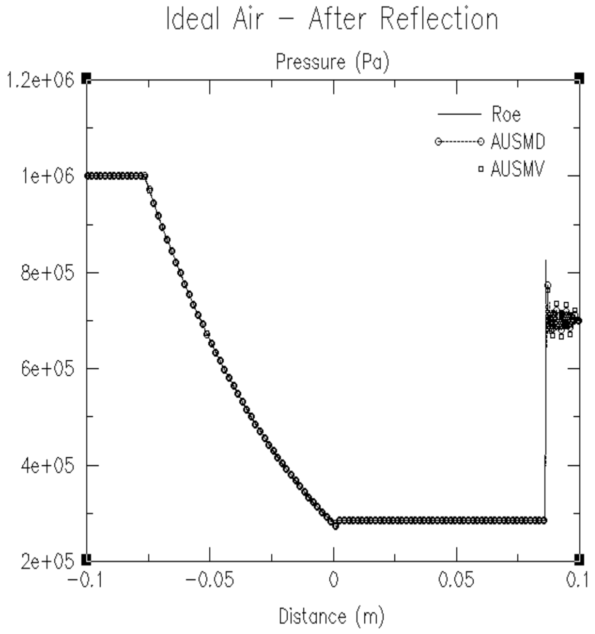


Figure 15

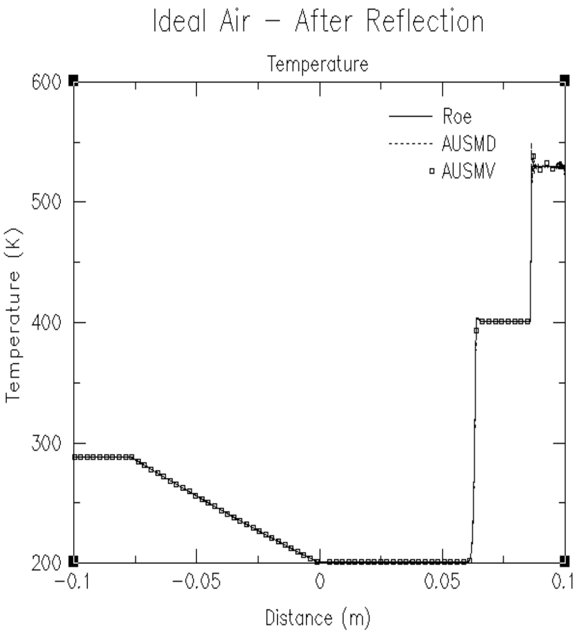


Figure 16

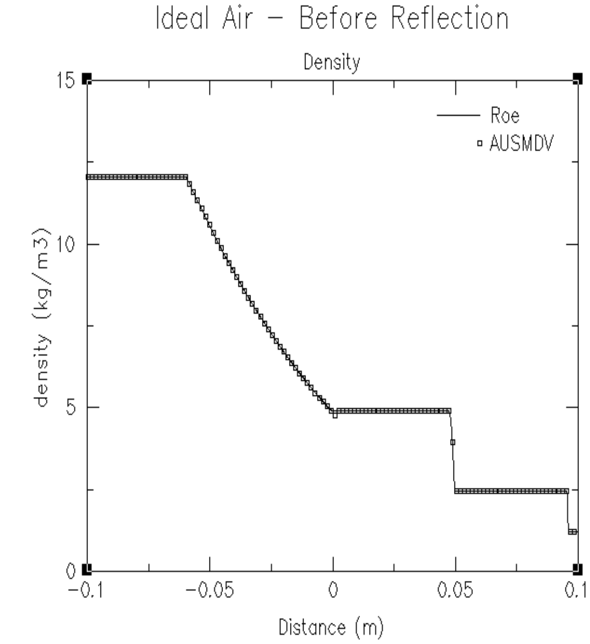


Figure 17

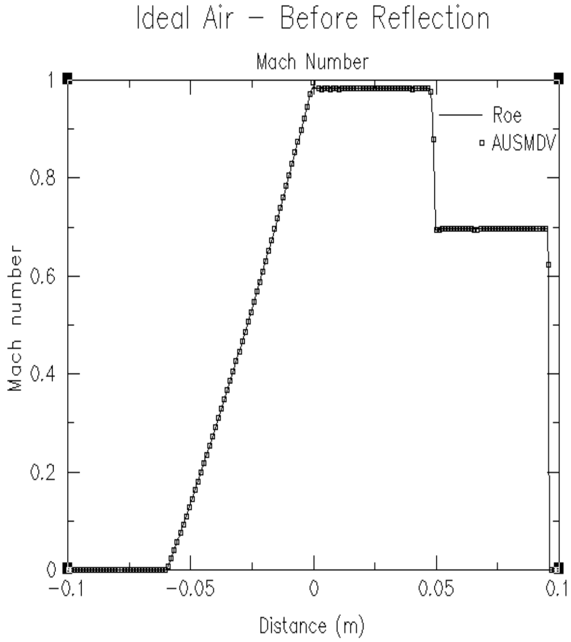


Figure 18

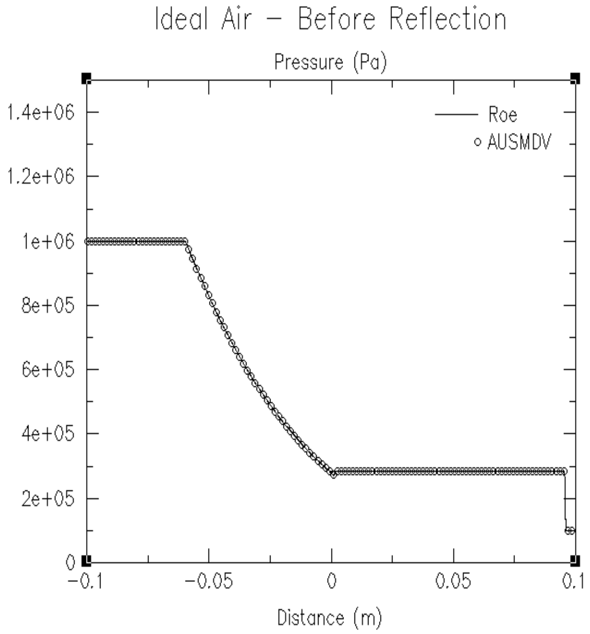


Figure 19

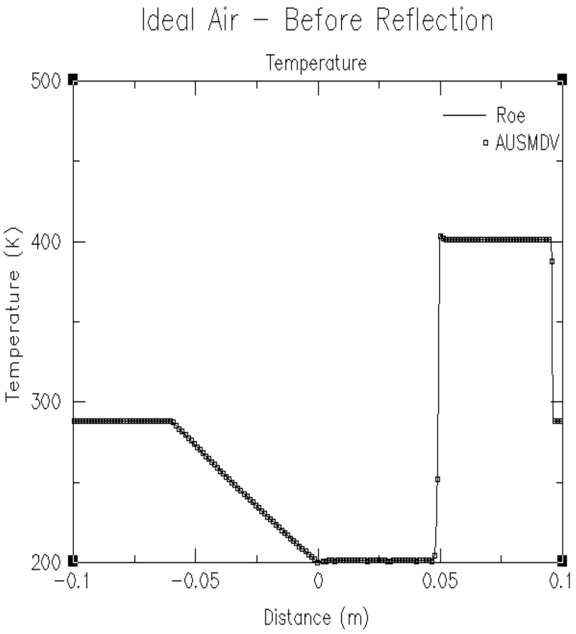


Figure 20

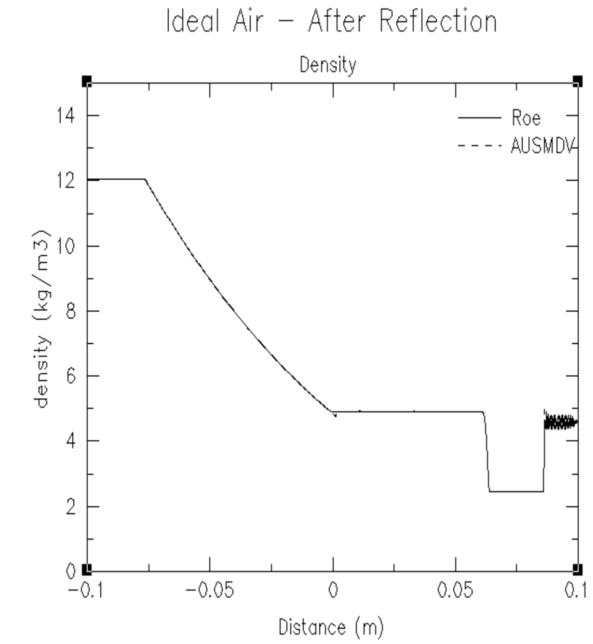


Figure 21

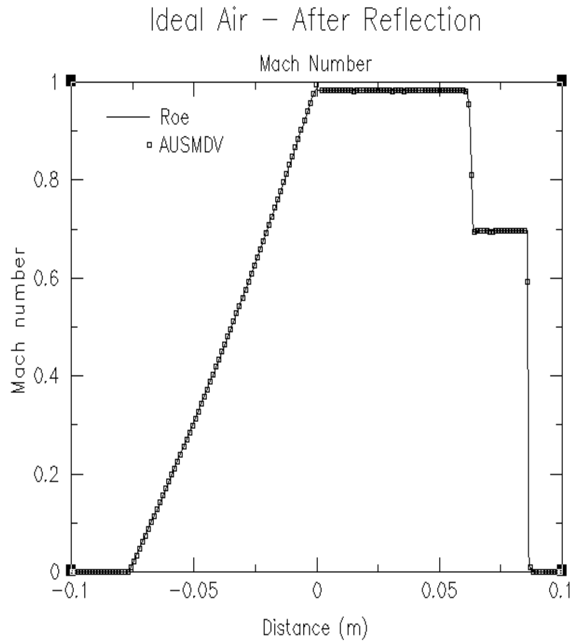


Figure 22

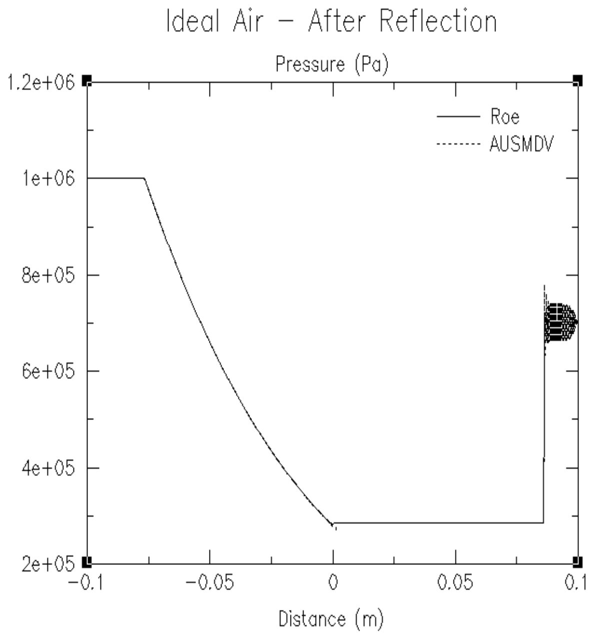


Figure 23

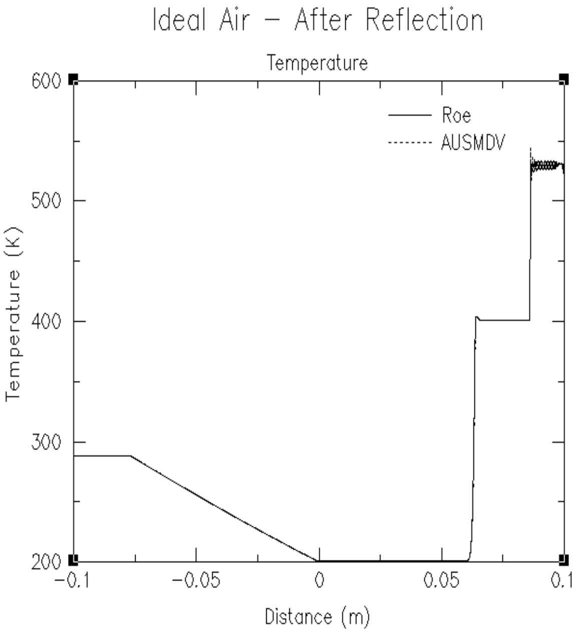


Figure 24

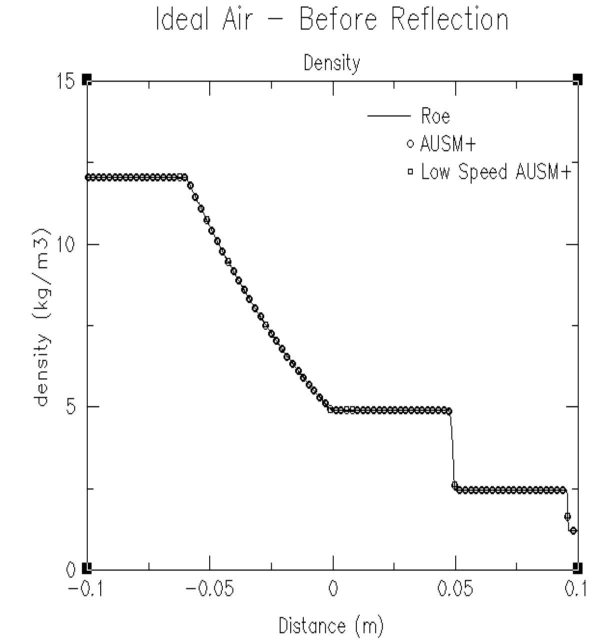


Figure 25

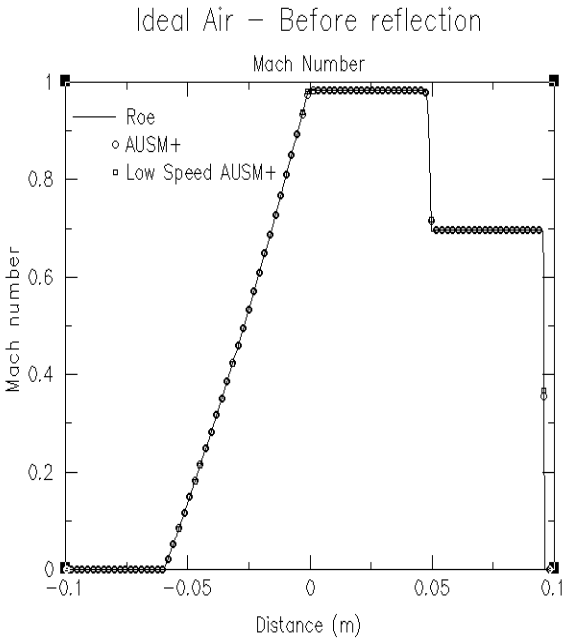


Figure 26

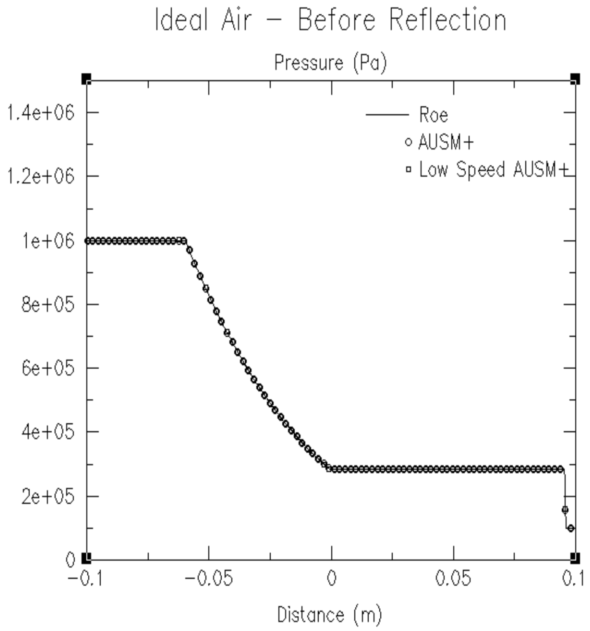


Figure 27

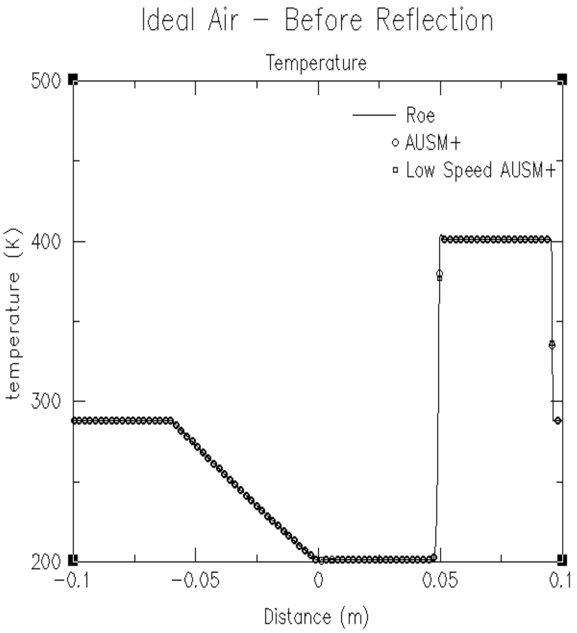


Figure 28

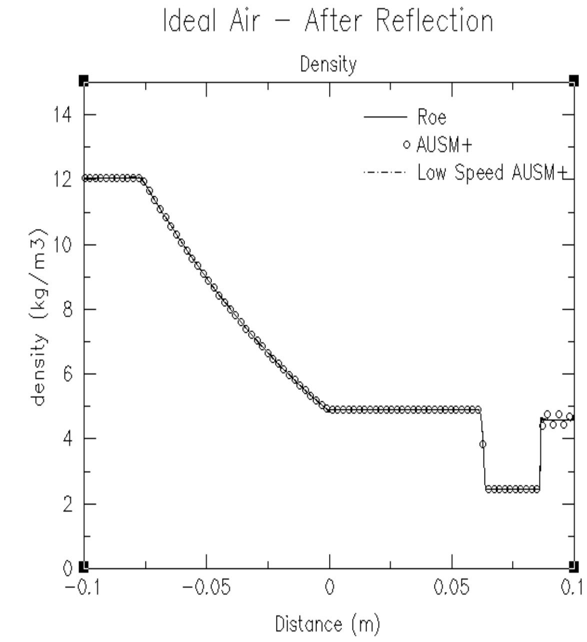


Figure 29

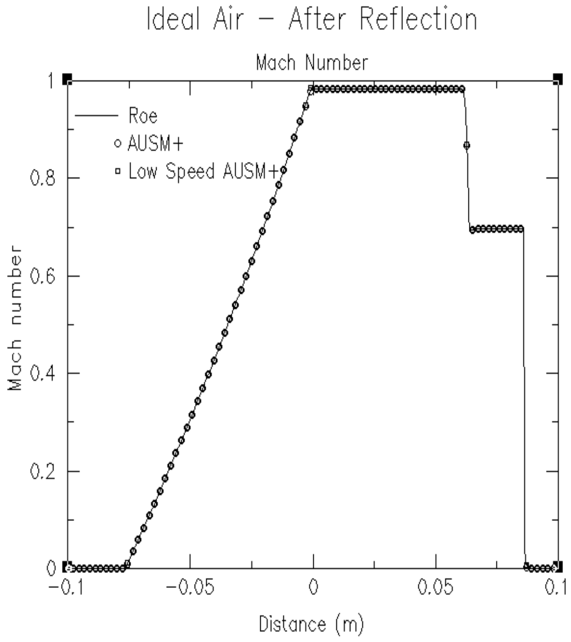


Figure 30

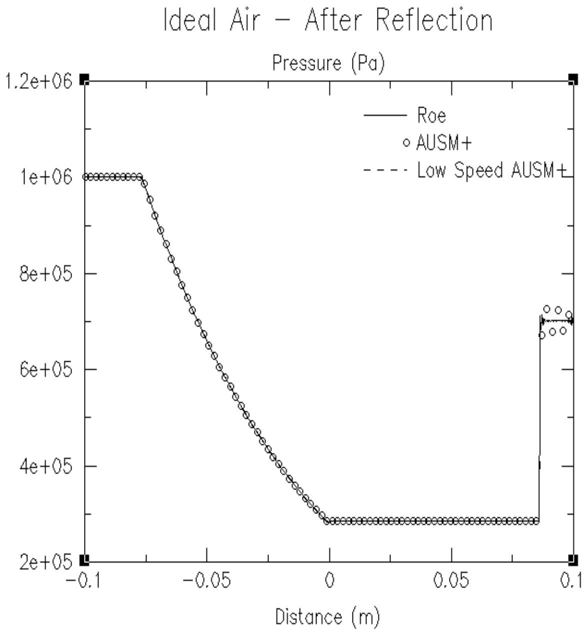


Figure 31

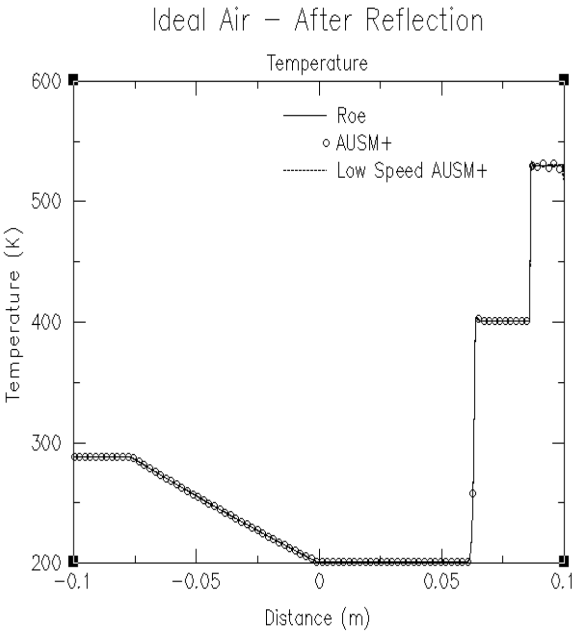


Figure 32

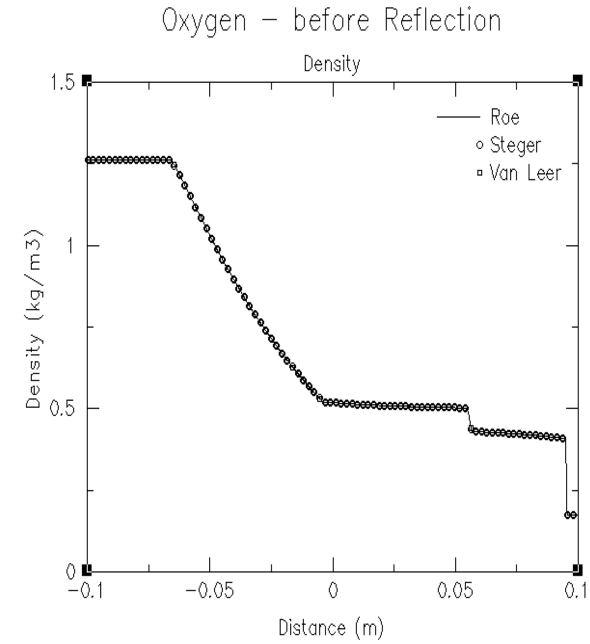


Figure 33

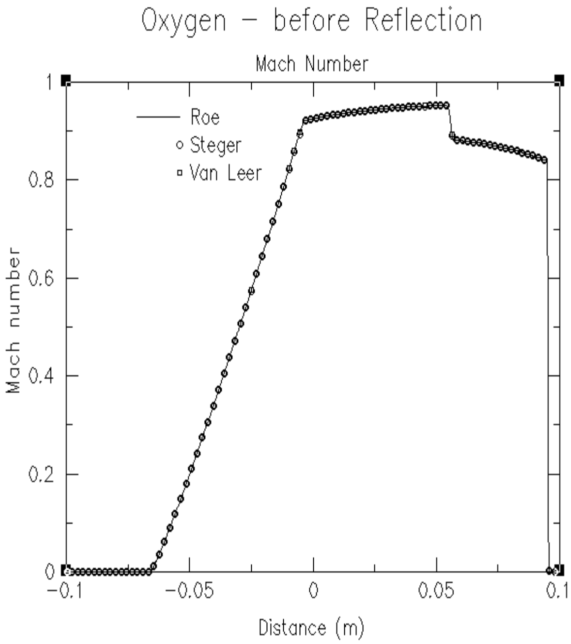


Figure 34

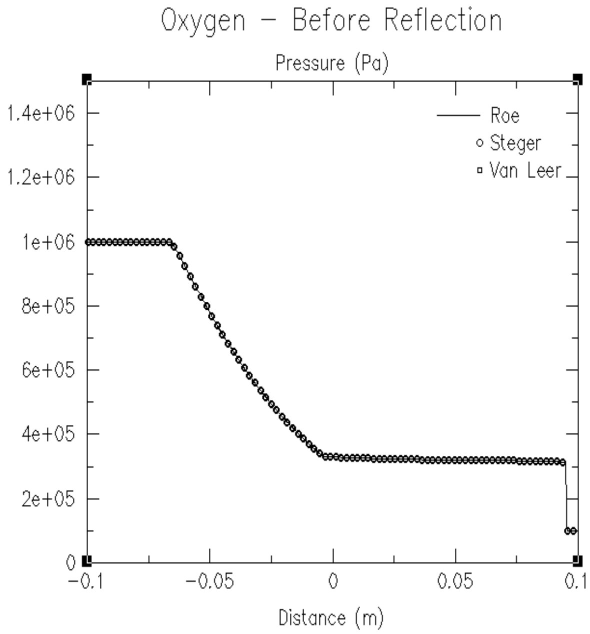


Figure 35

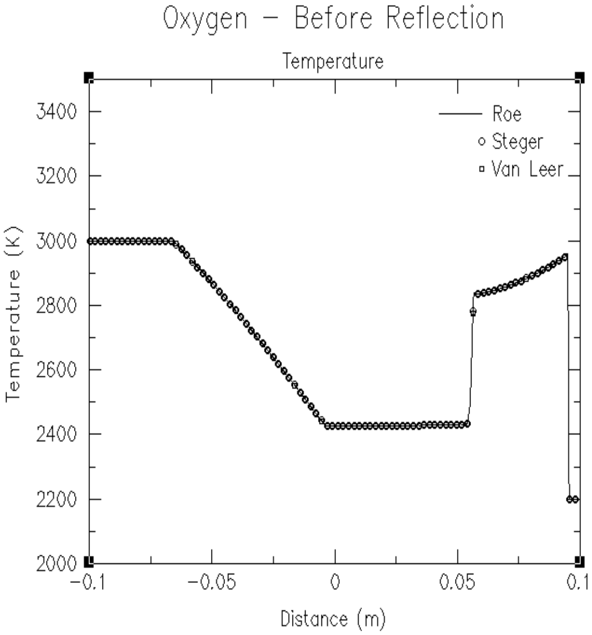


Figure 36

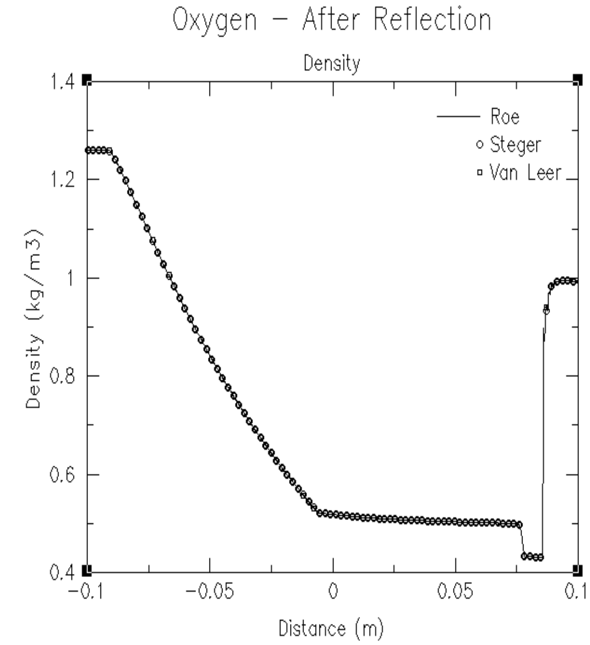


Figure 37

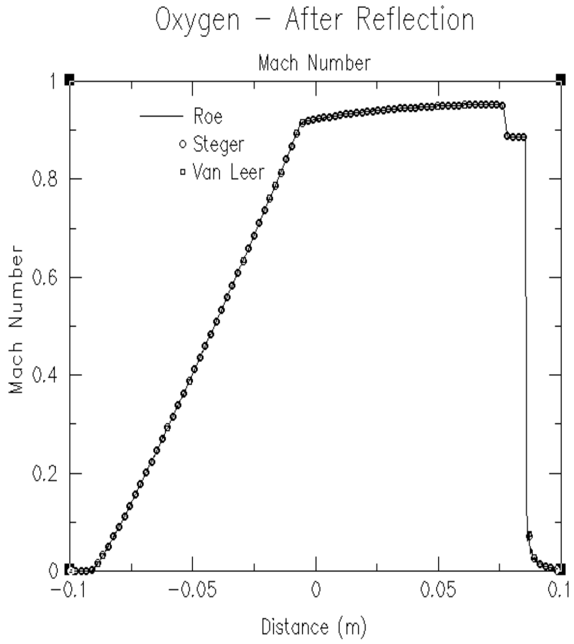


Figure 38

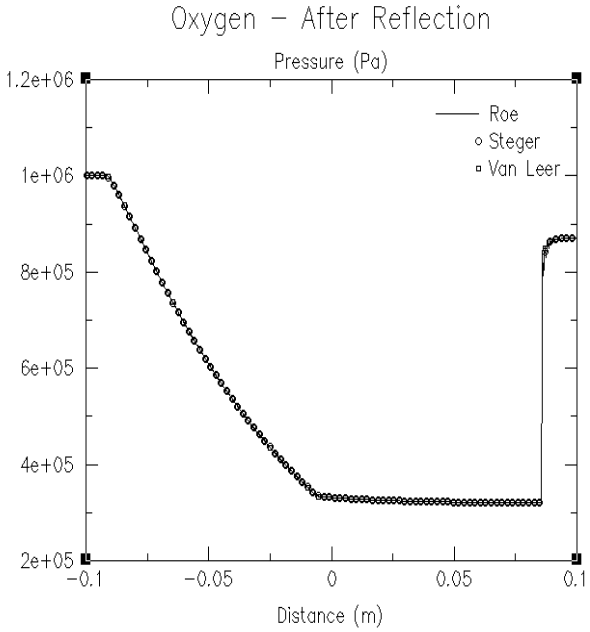


Figure 39

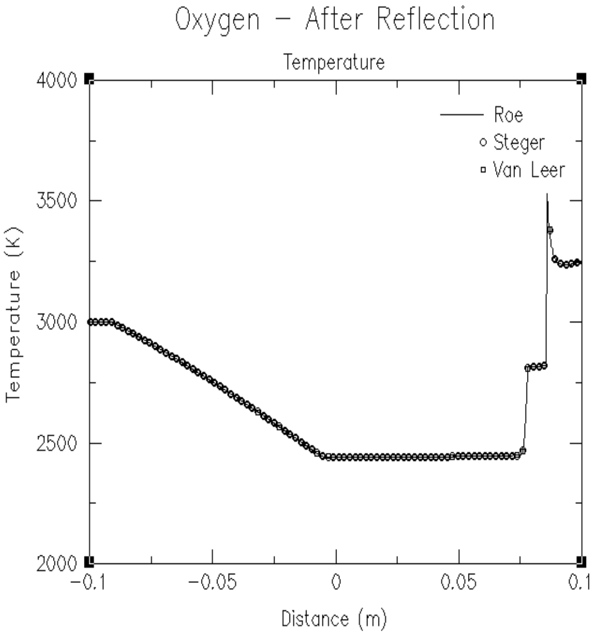


Figure 40

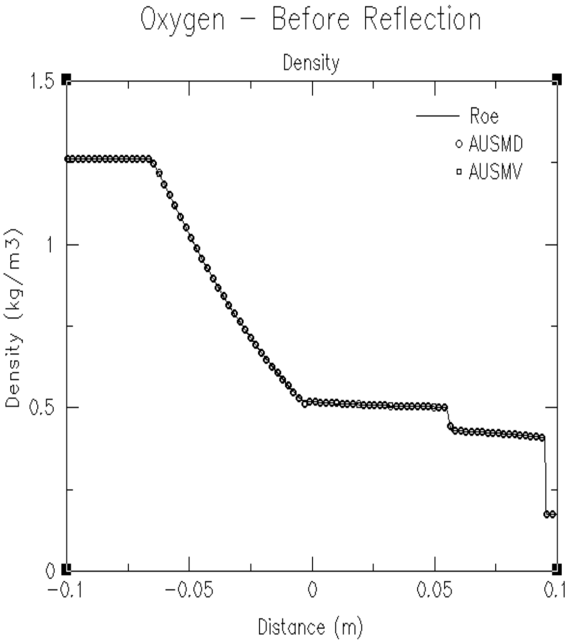


Figure 41

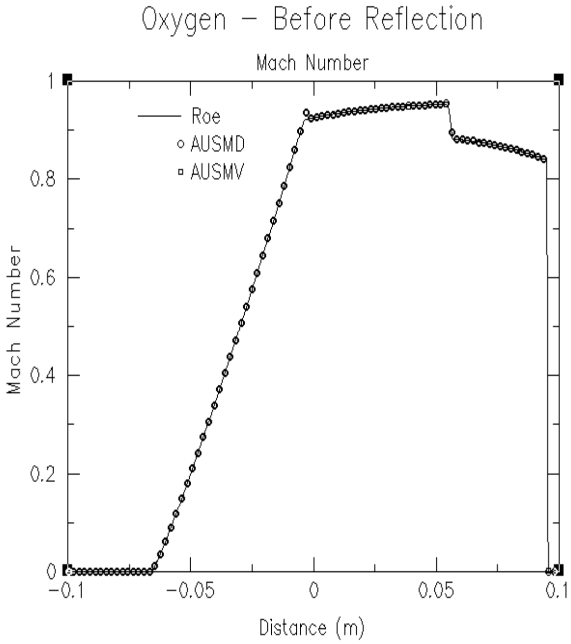


Figure 42

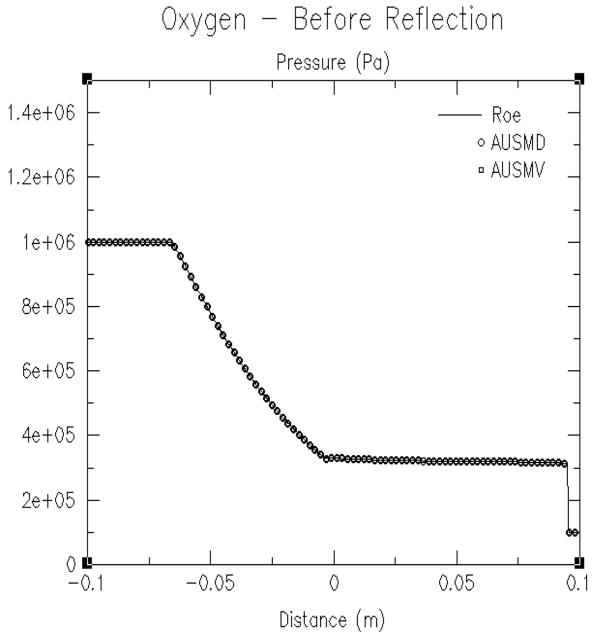


Figure 43

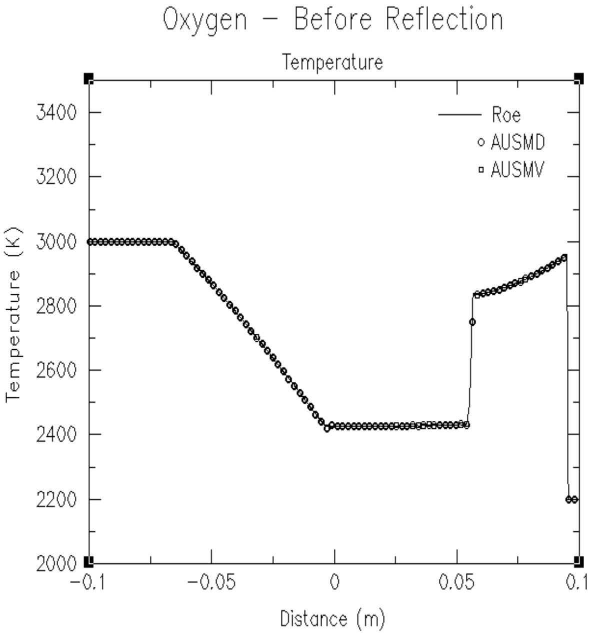


Figure 44

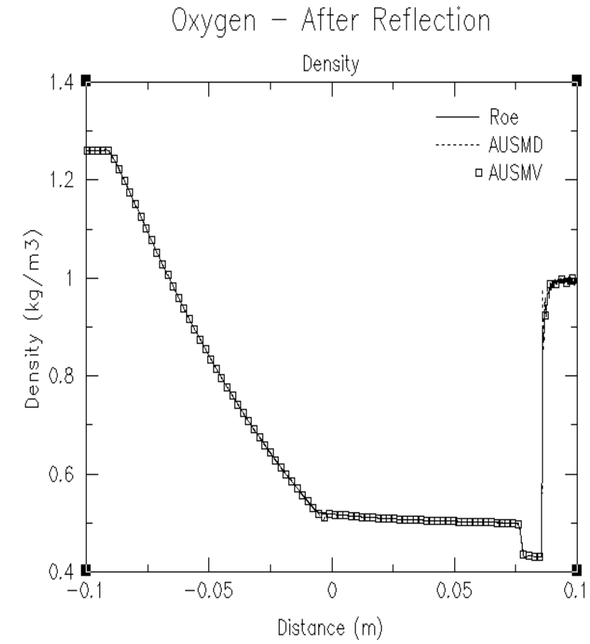


Figure 45

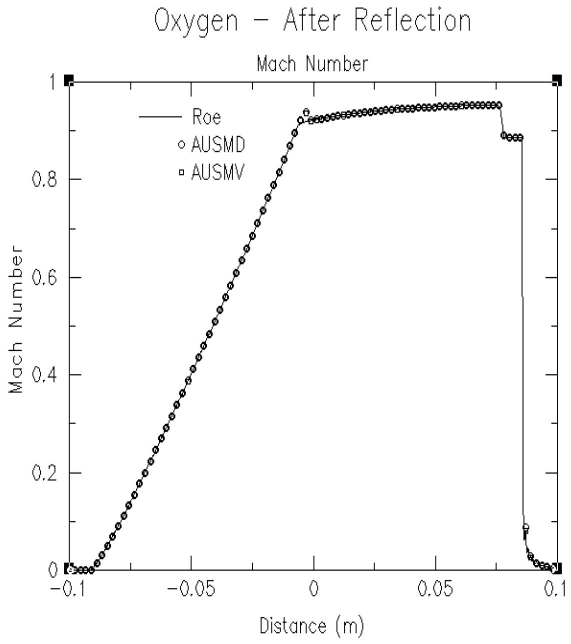


Figure 46

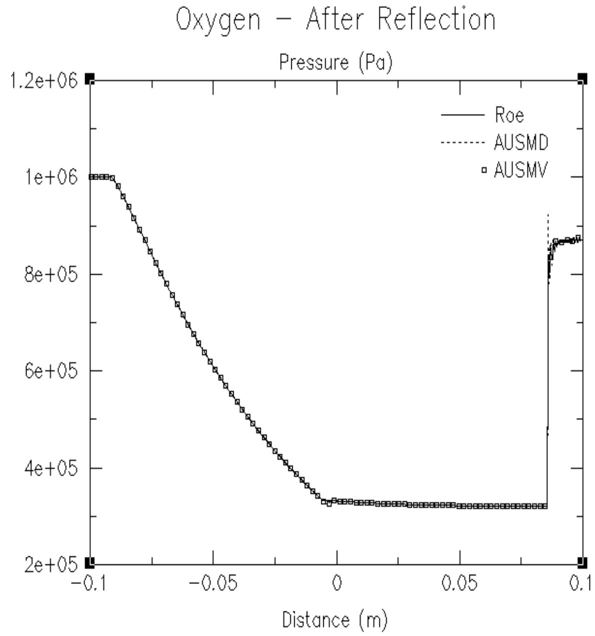


Figure 47

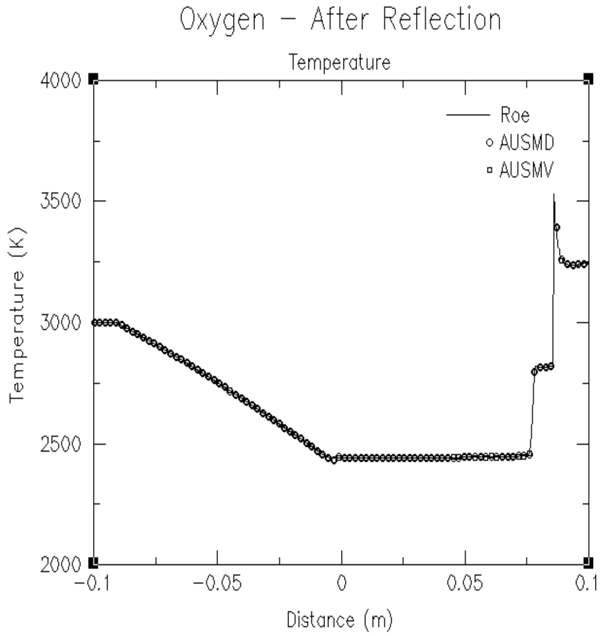


Figure 48

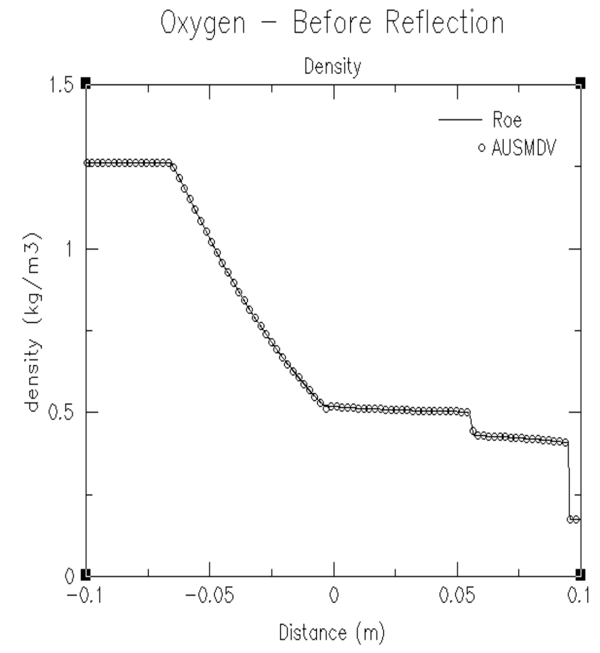


Figure 49

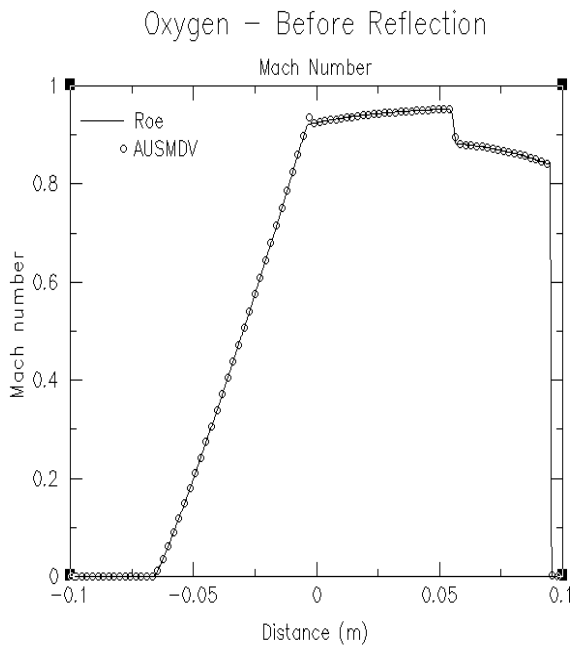


Figure 50

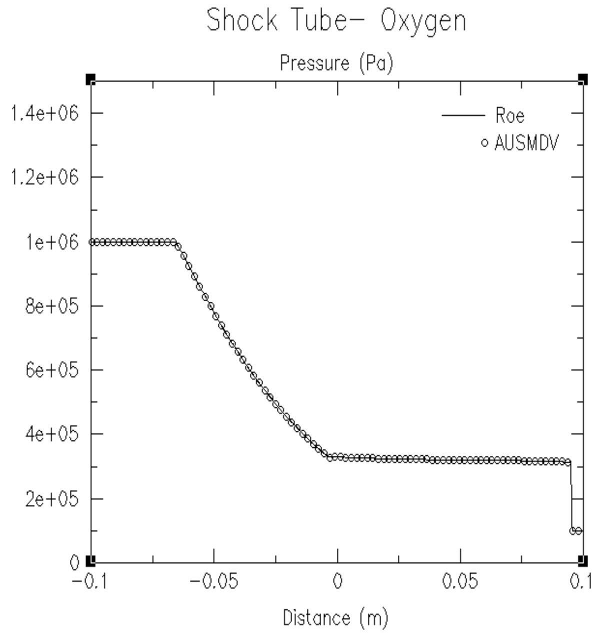


Figure 51

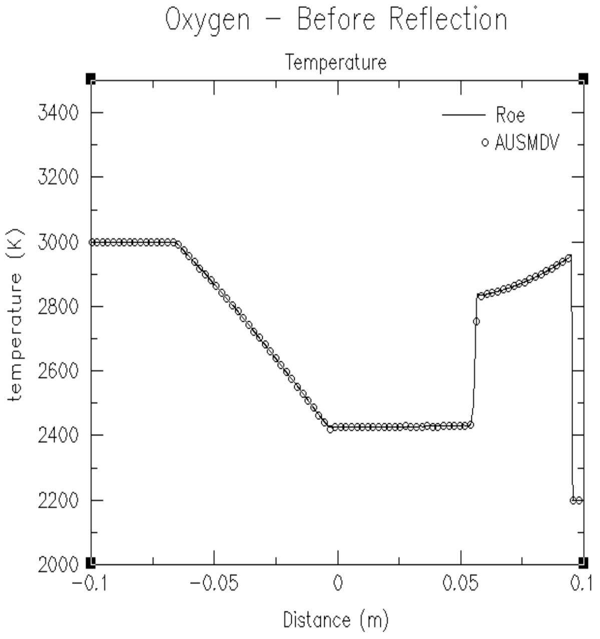


Figure 52

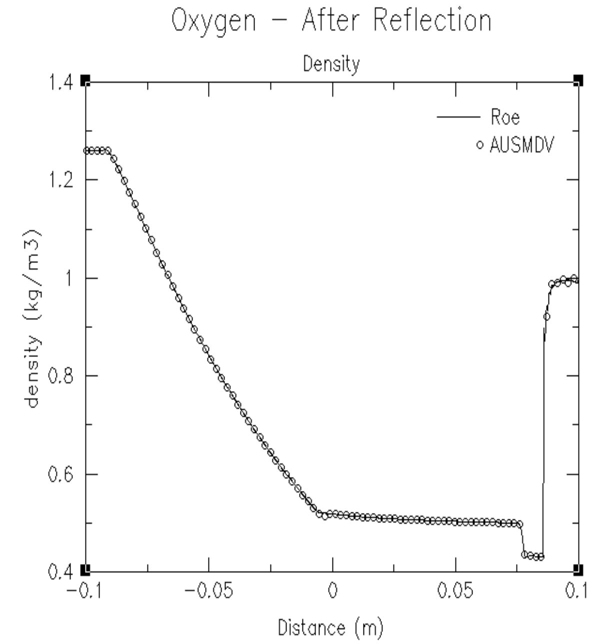


Figure 53

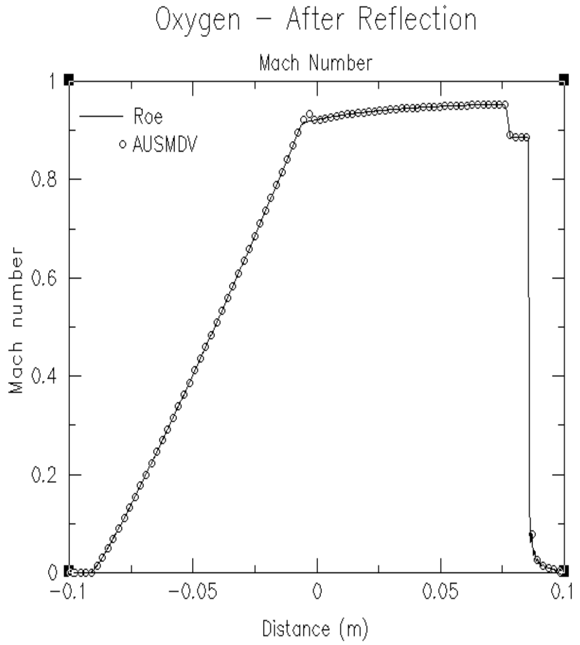


Figure 54

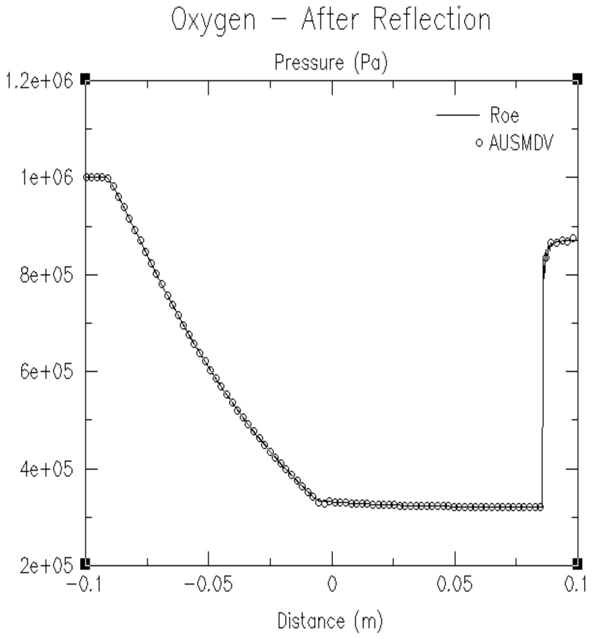


Figure 55

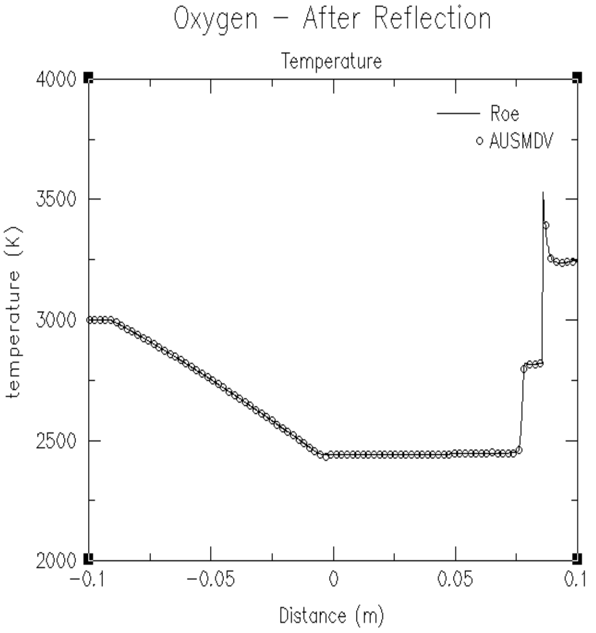


Figure 56

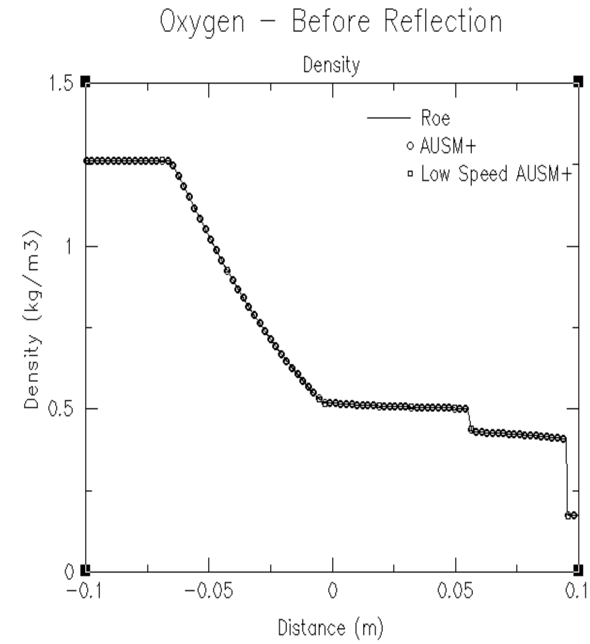


Figure 57

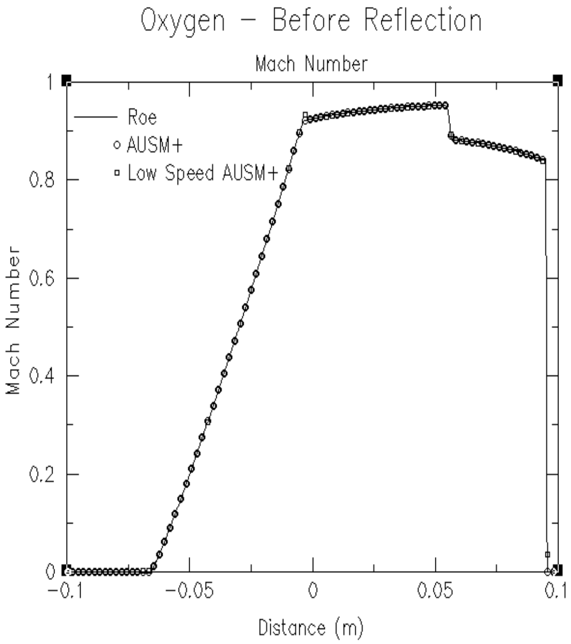


Figure 58

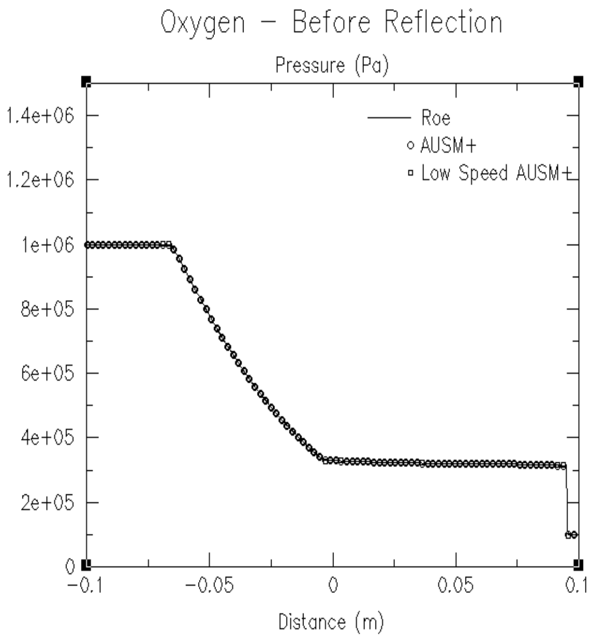


Figure 59

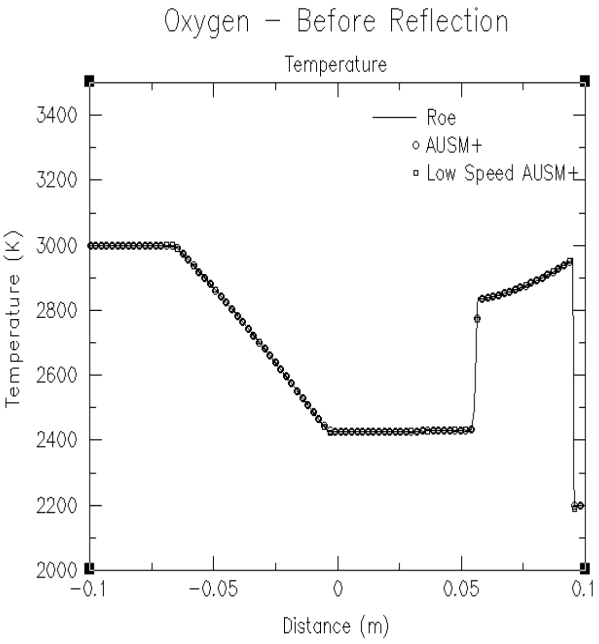


Figure 60

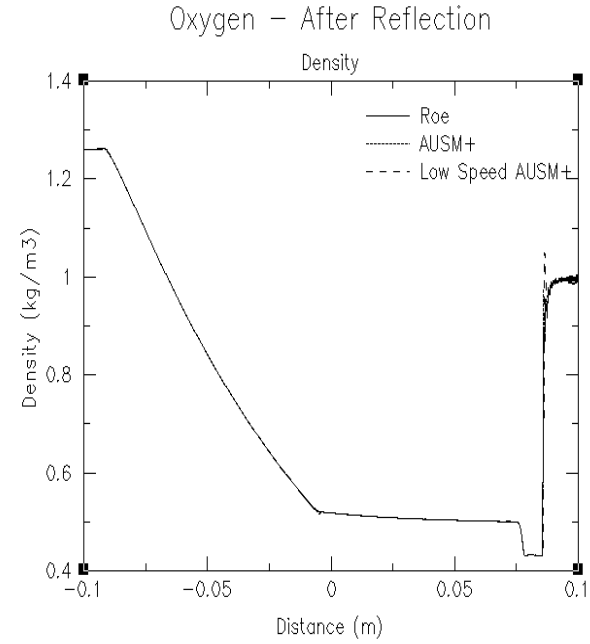


Figure 61

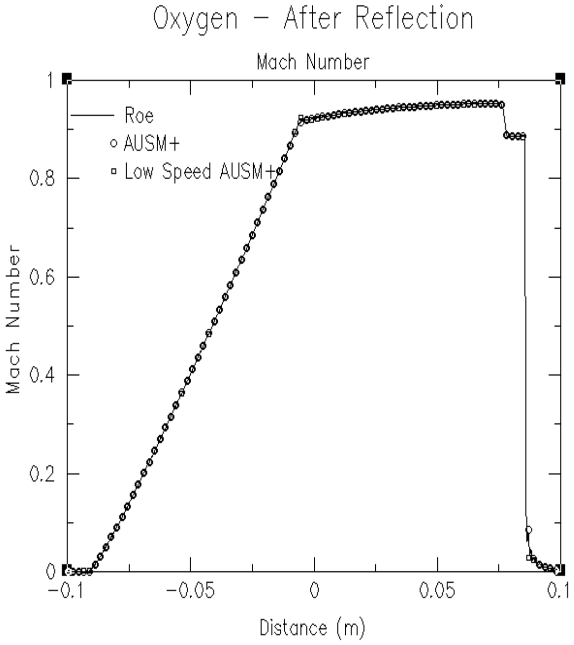


Figure 62

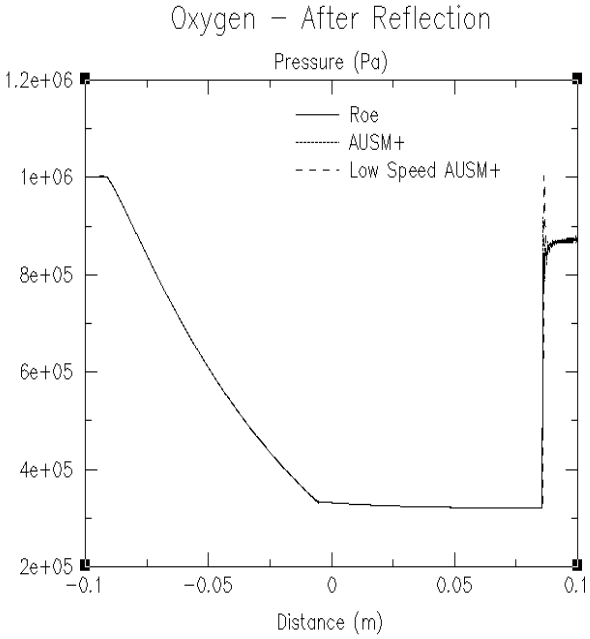


Figure 63

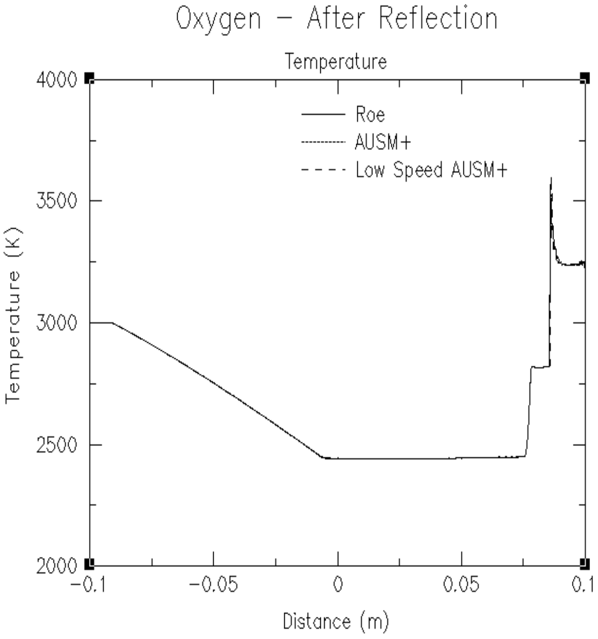


Figure 64

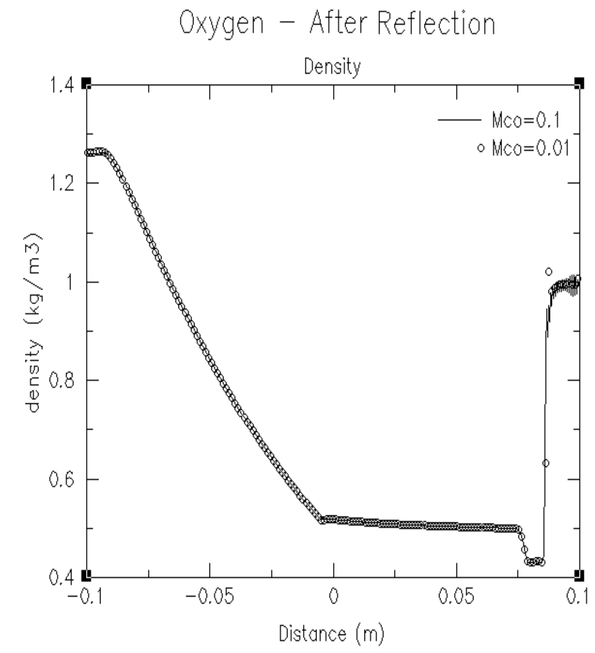


Figure 65

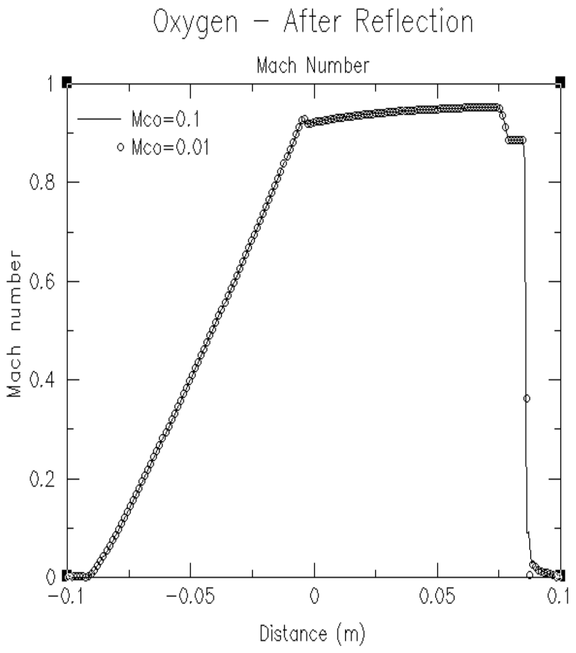


Figure 66

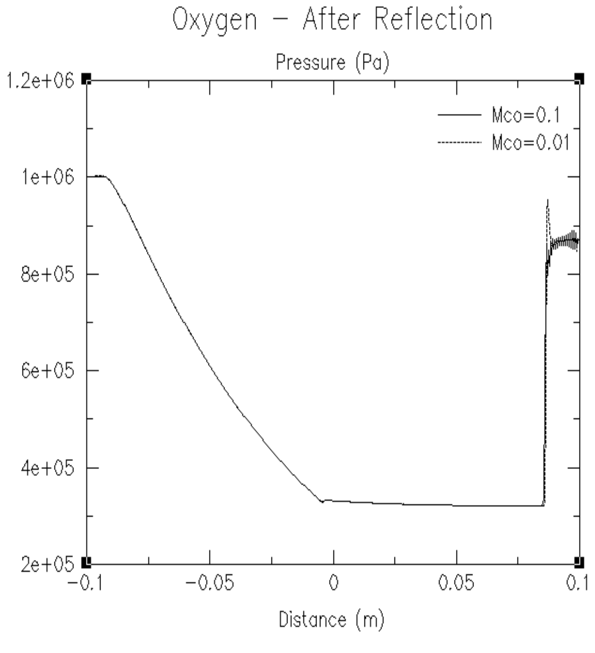


Figure 67

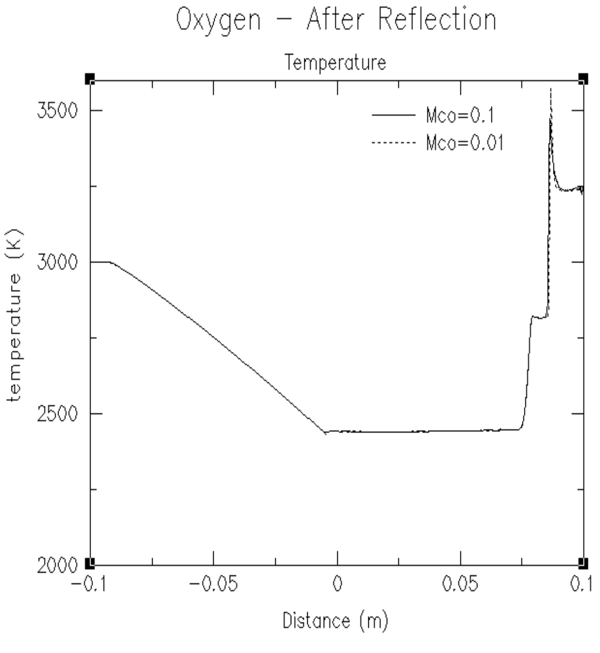


Figure 68

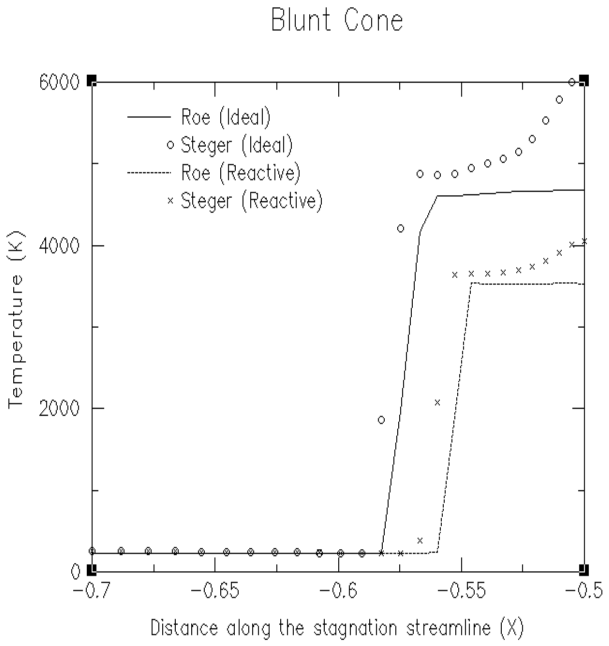


Figure 69

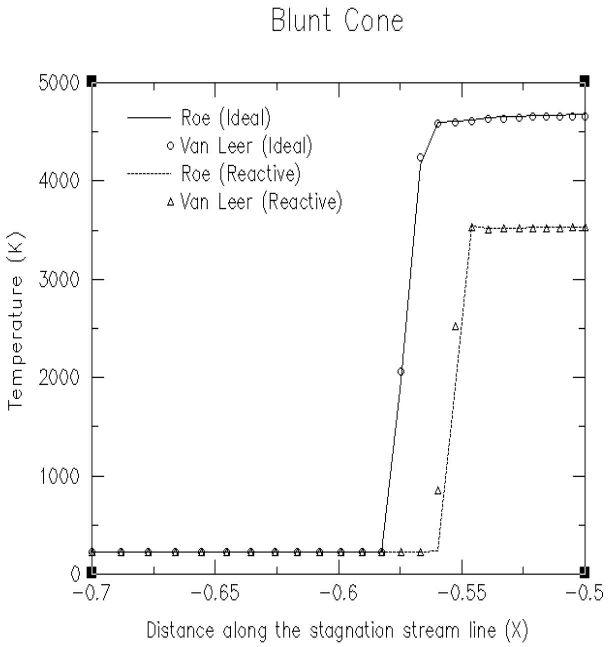


Figure 70

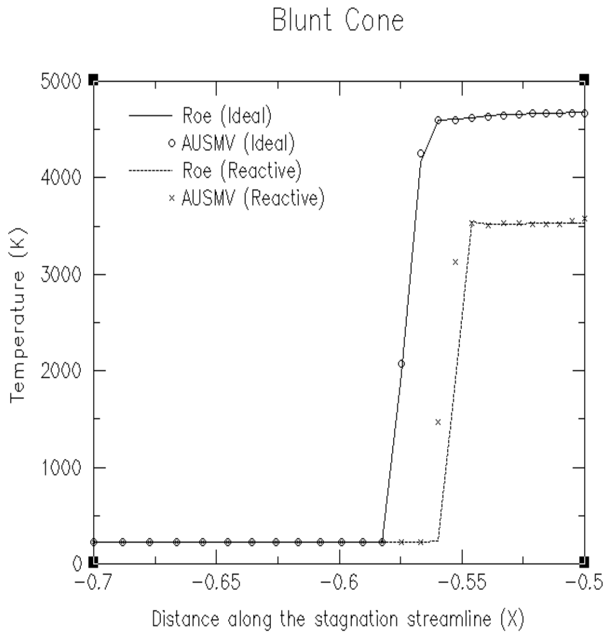


Figure 71

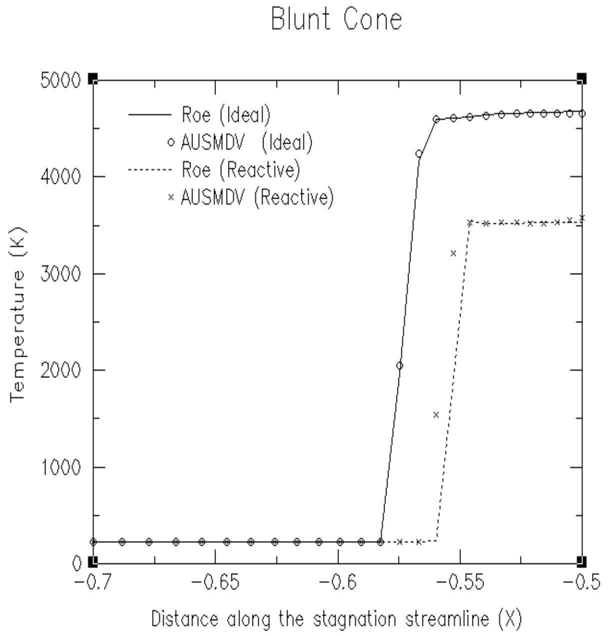


Figure 72

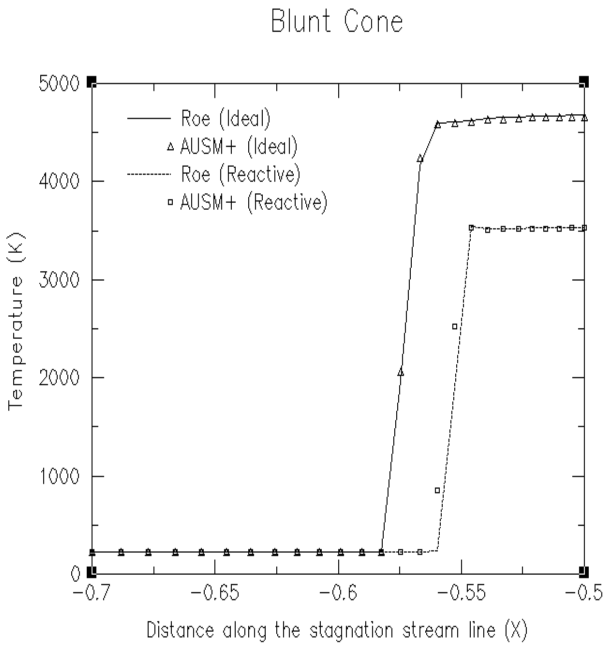


Figure 73

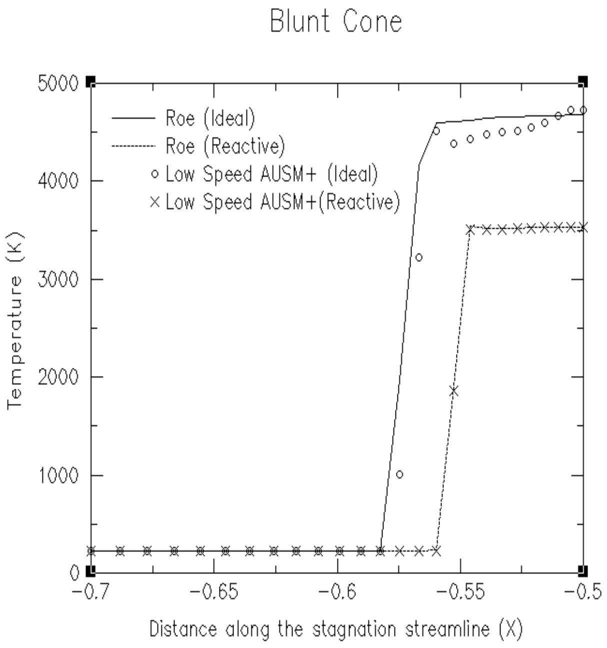


Figure 74

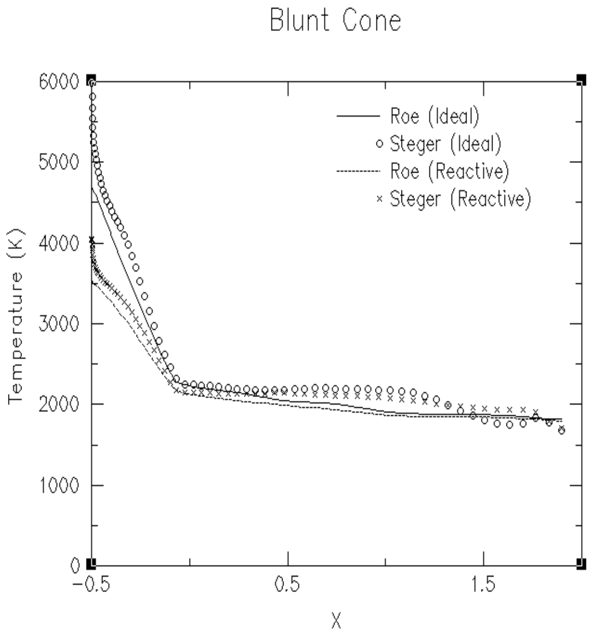


Figure 75

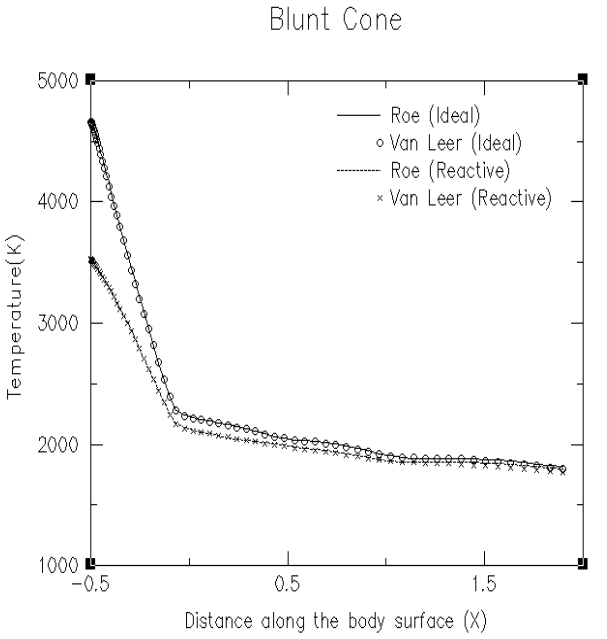


Figure 76

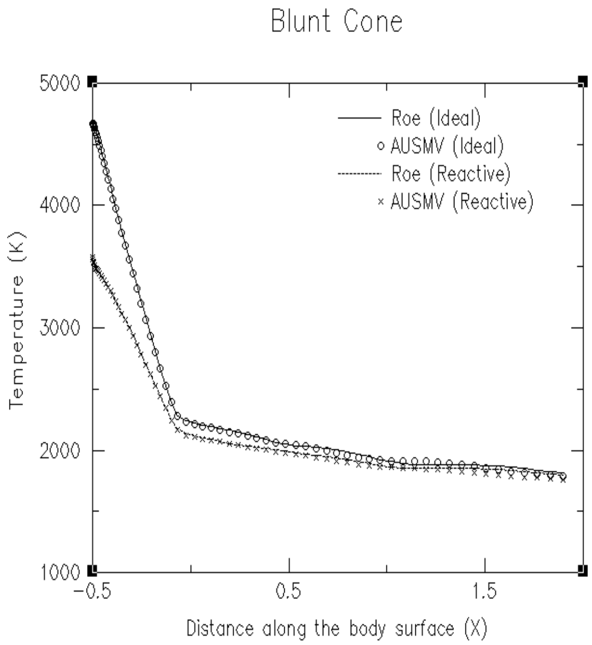


Figure 77

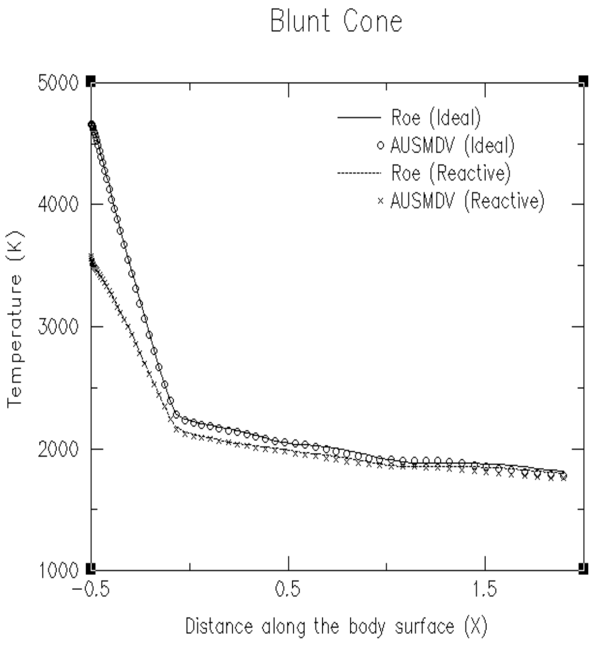
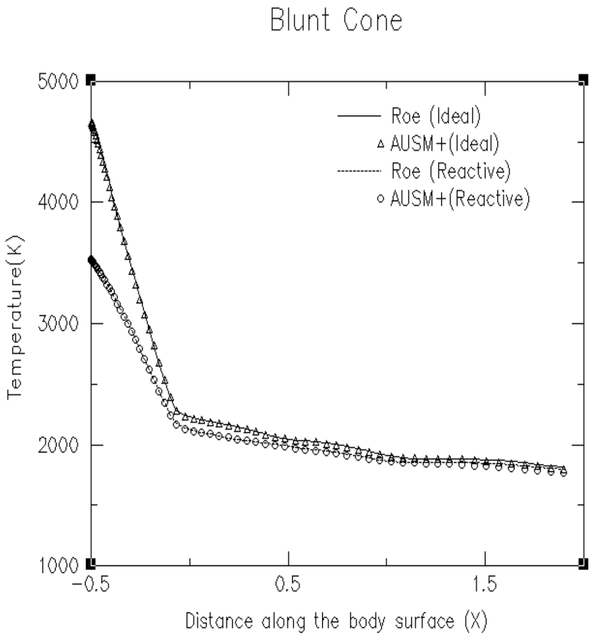


Figure 78



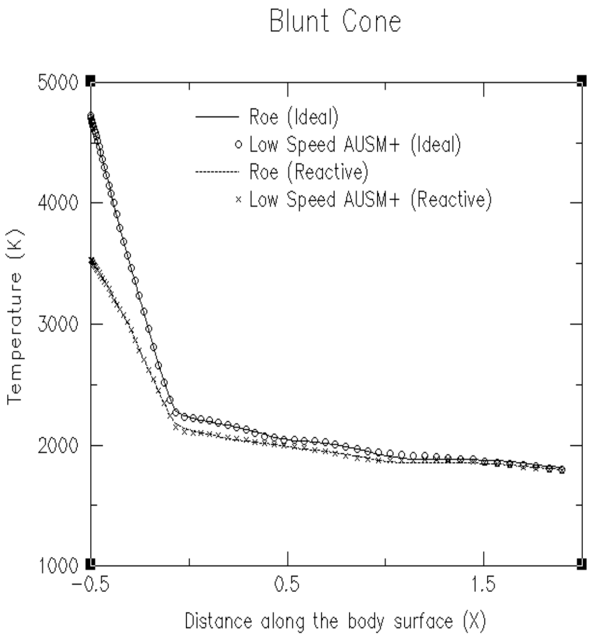
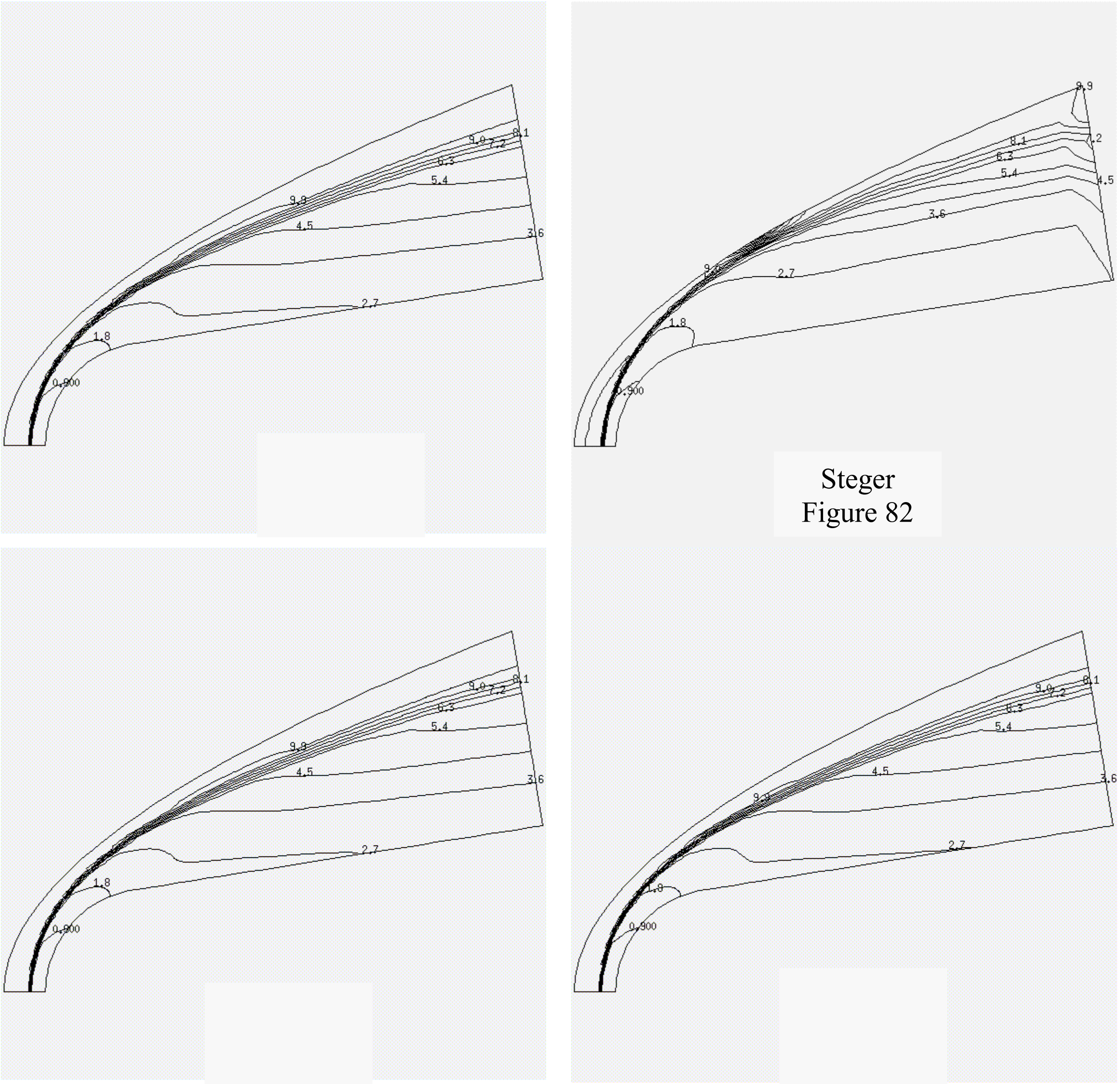
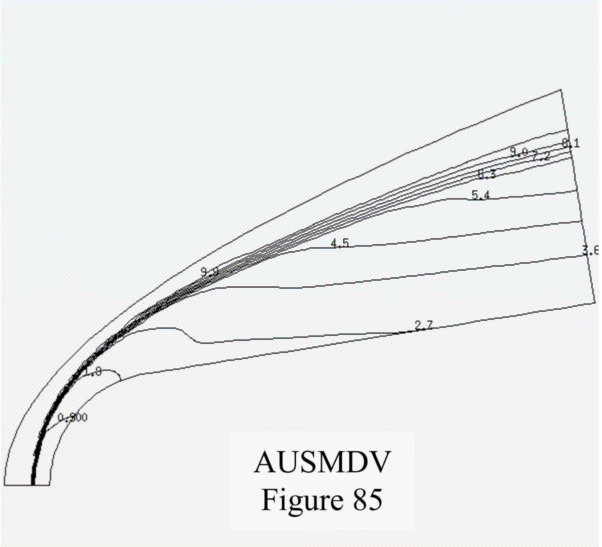


Figure 80

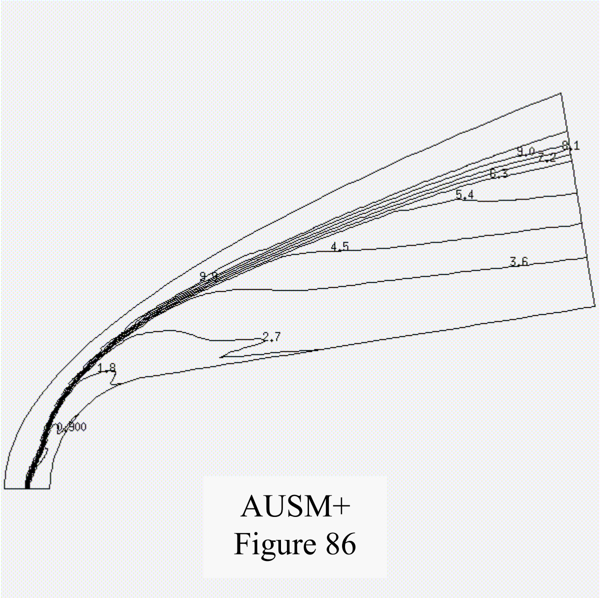
Mach Number contours for Ideal Gas Model



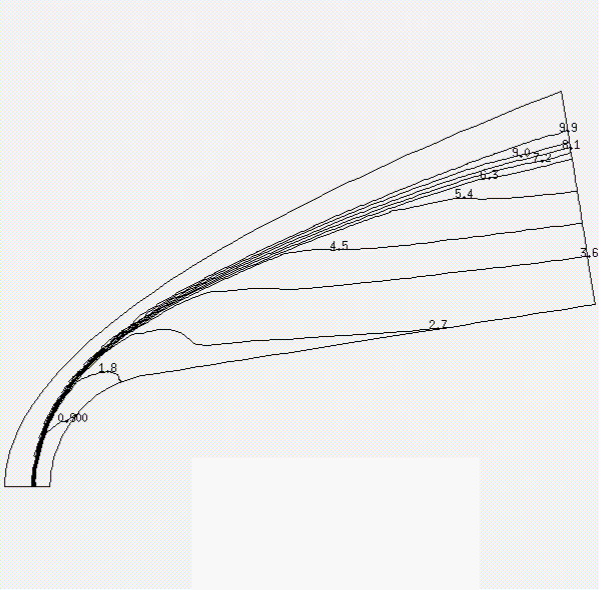
Steger
Figure 82



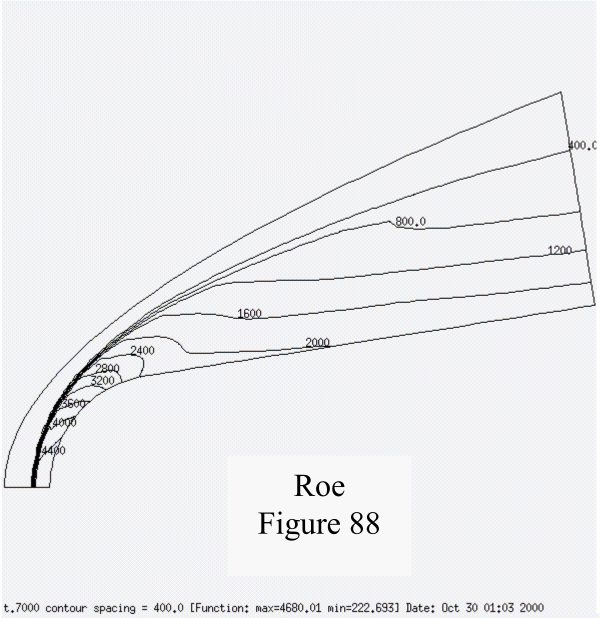
AUSMDV
Figure 85



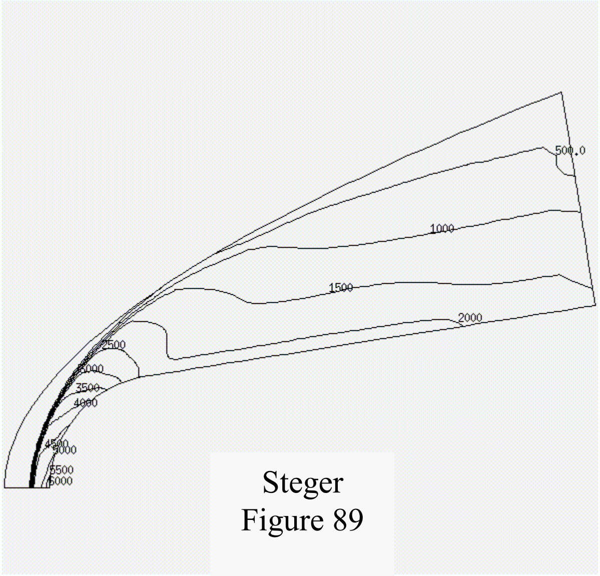
AUSM+
Figure 86



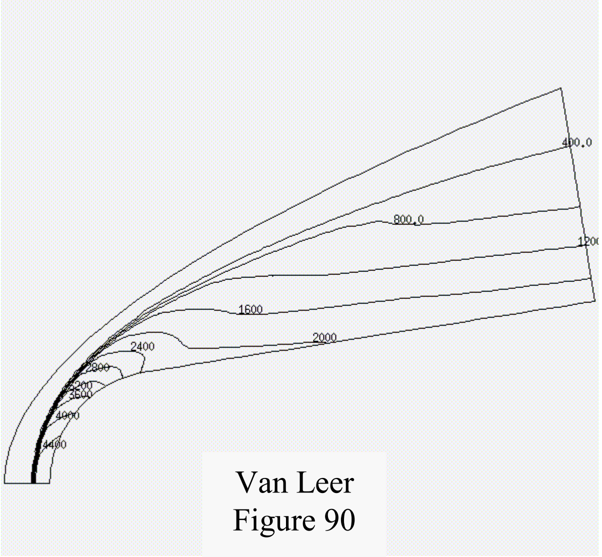
Temperature contours for Ideal Gas Model (in degrees K)



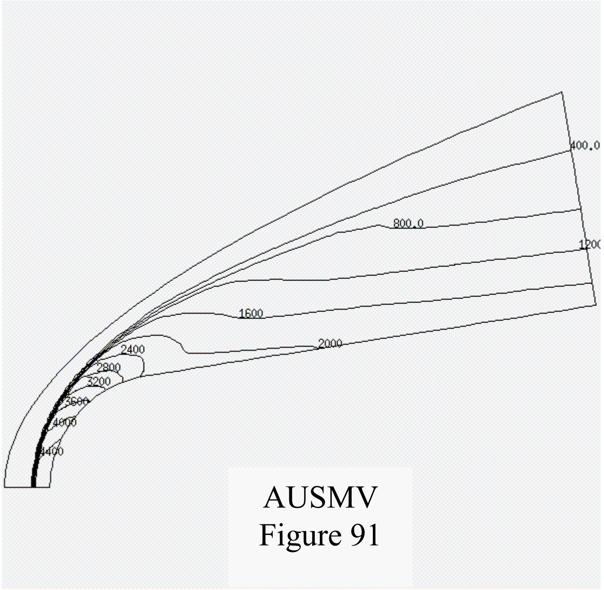
Roe
Figure 88



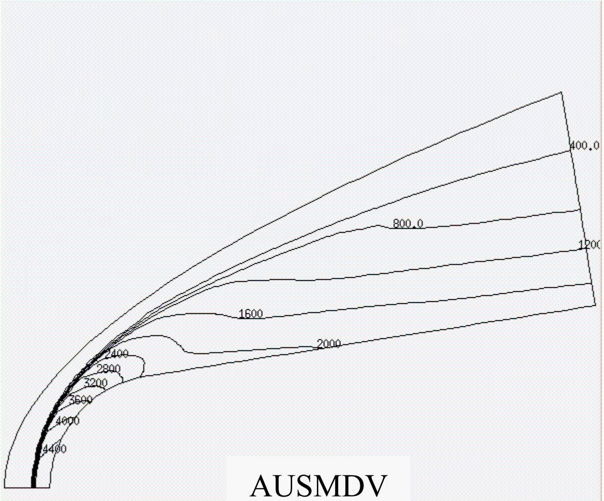
Steger
Figure 89



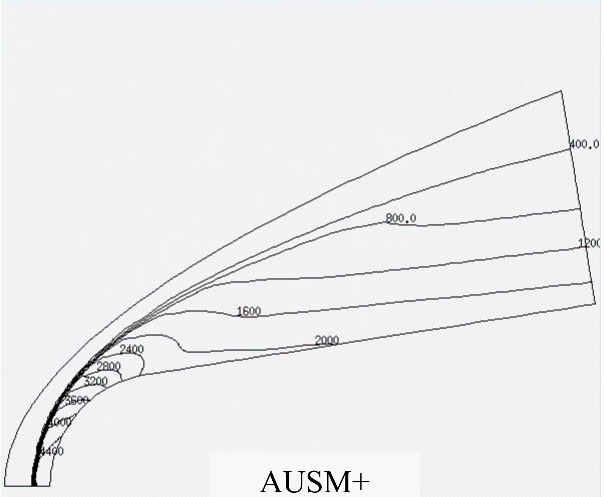
Van Leer
Figure 90



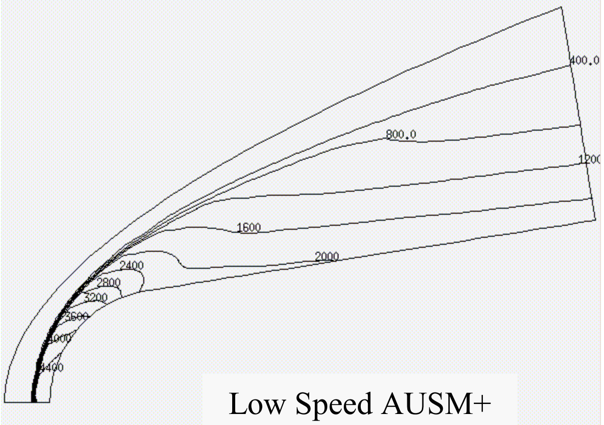
AUSMV
Figure 91



AUSMDV
Figure 92

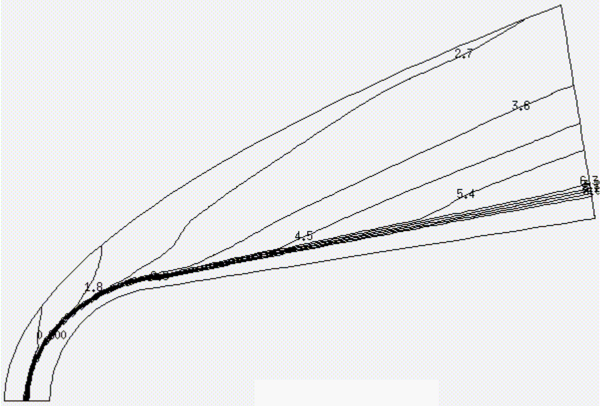


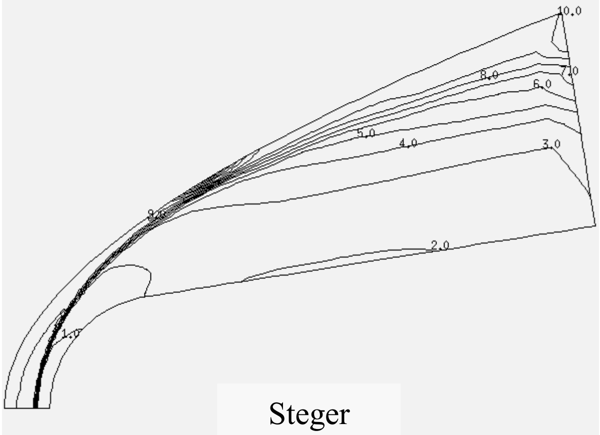
AUSM+
Figure 93



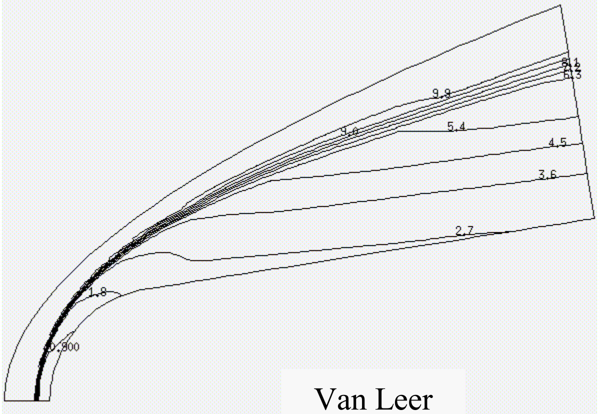
Low Speed AUSM+
Figure 94

Mach Number contours for Chemically Reacting Air

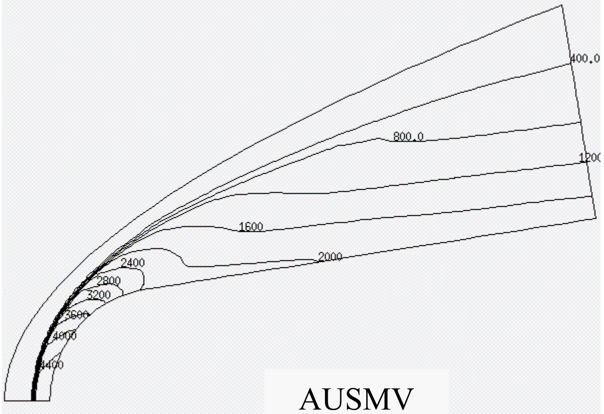




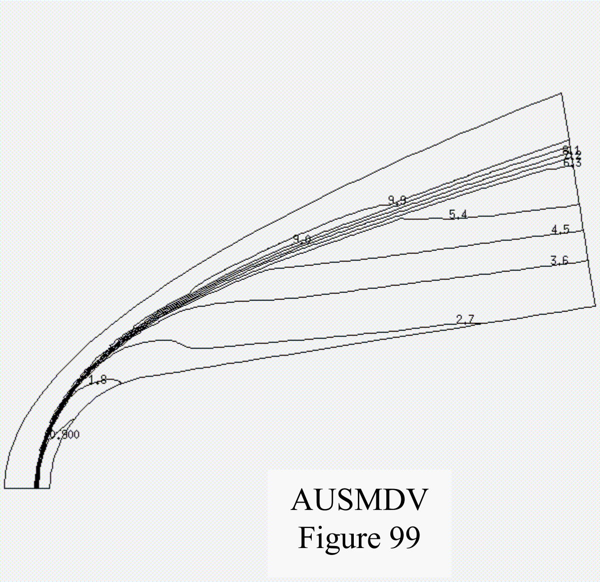
Steger
Figure 96



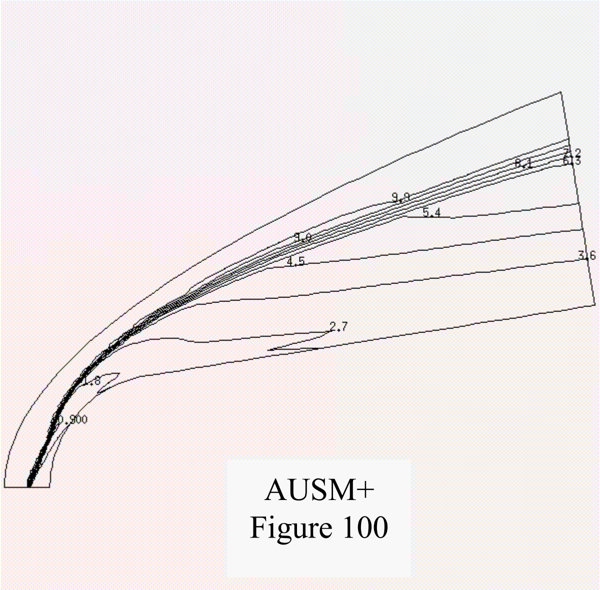
Van Leer
Figure 97



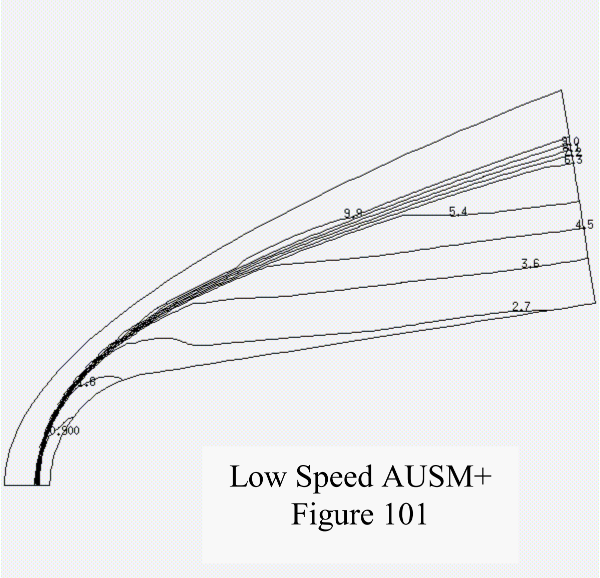
AUSMV
Figure 98



AUSMDV
Figure 99

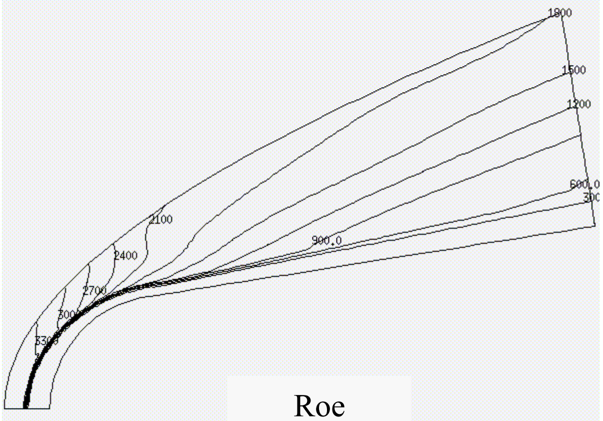


AUSM+
Figure 100

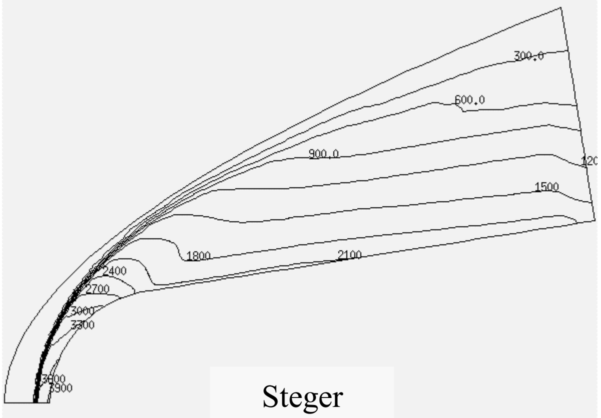


Low Speed AUSM+
Figure 101

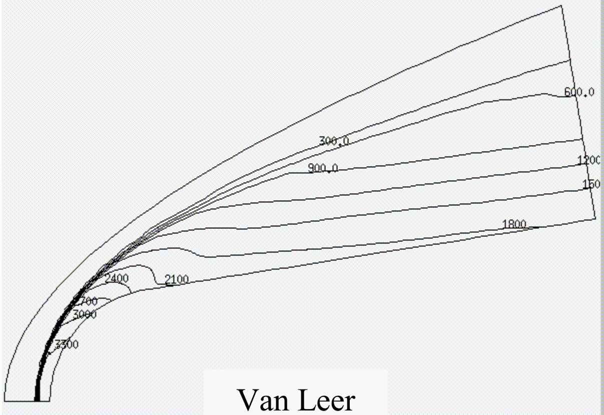
Temperature contours for Chemically Reacting Air (in degrees K)



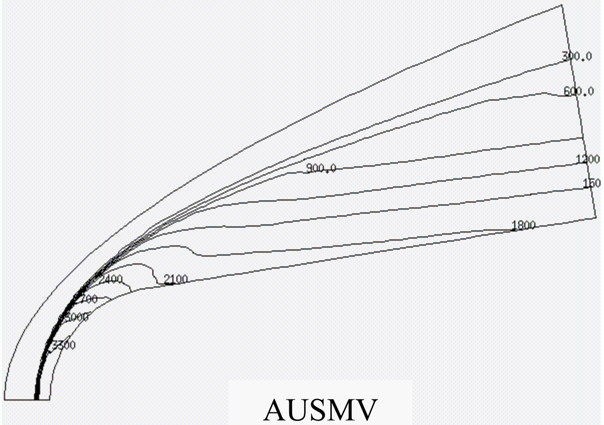
Roe
Figure 102



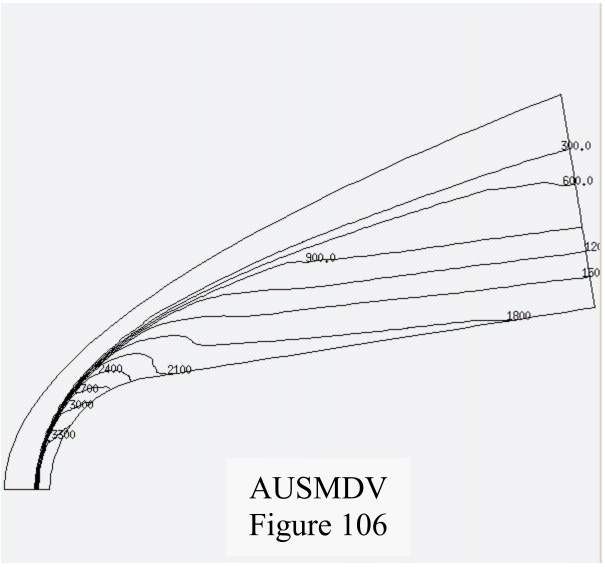
Steger
Figure 103



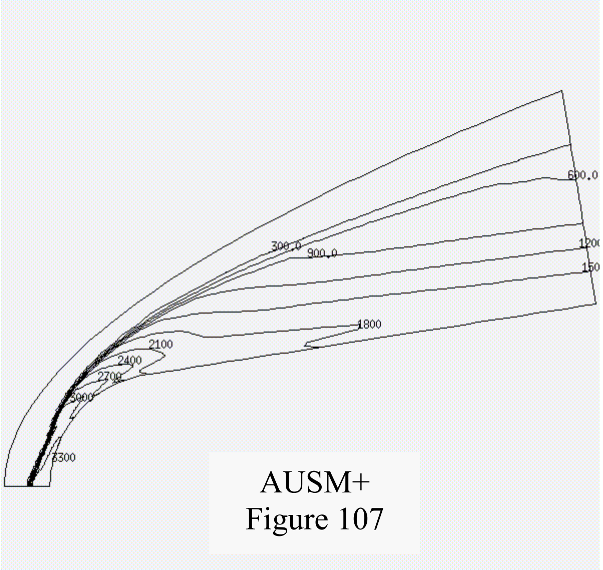
Van Leer
Figure 104



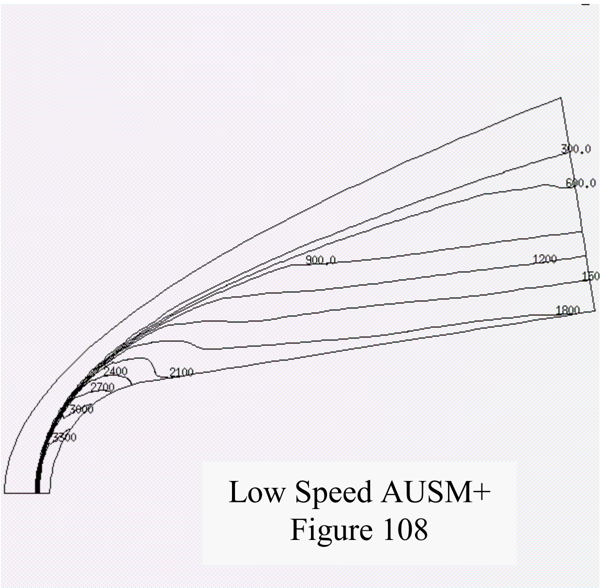
AUSMV
Figure 105



AUSMDV
Figure 106



AUSM+
Figure 107



Low Speed AUSM+
Figure 108

2D Impingement

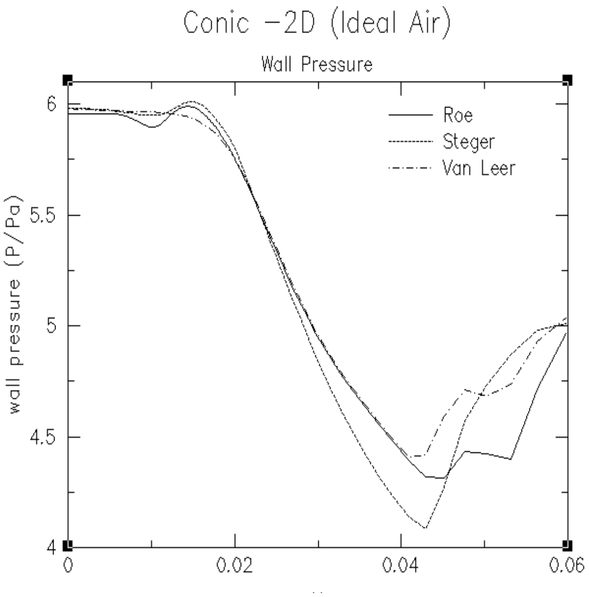


Figure 109

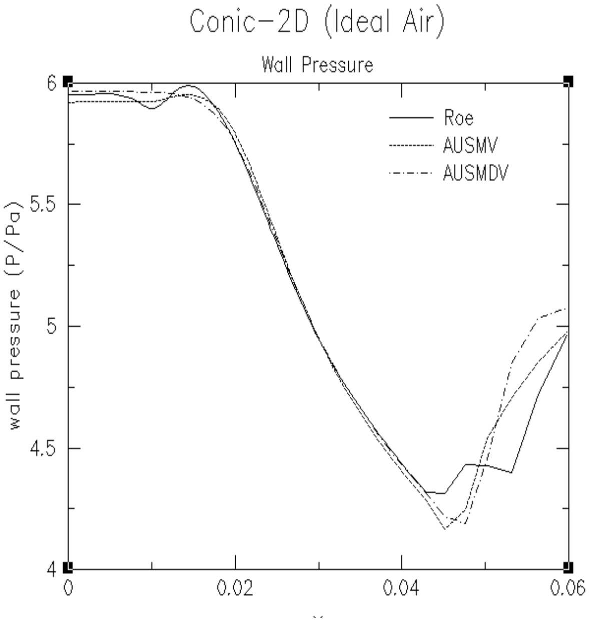


Figure 110

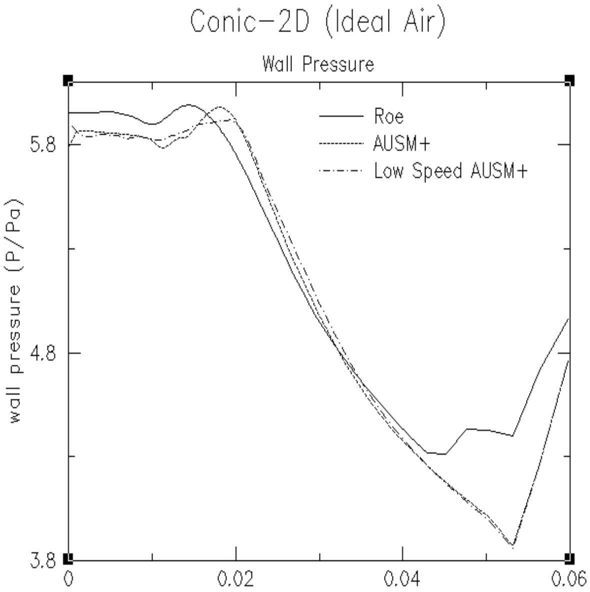


Figure 111

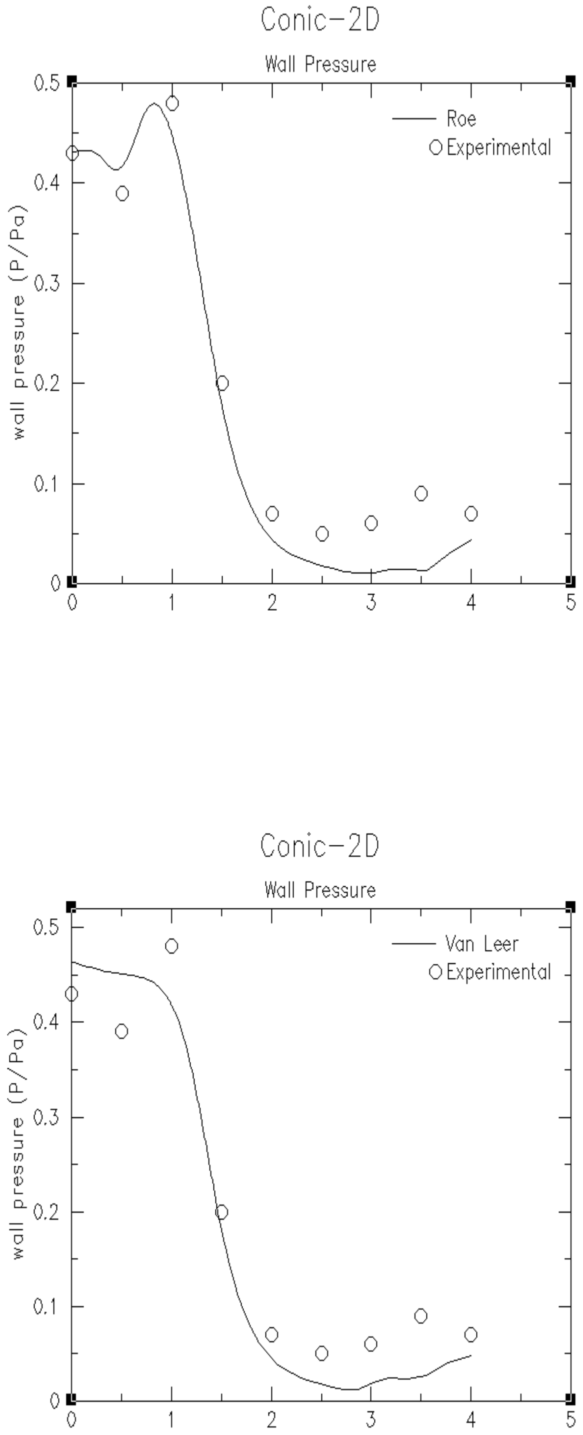


Figure 114

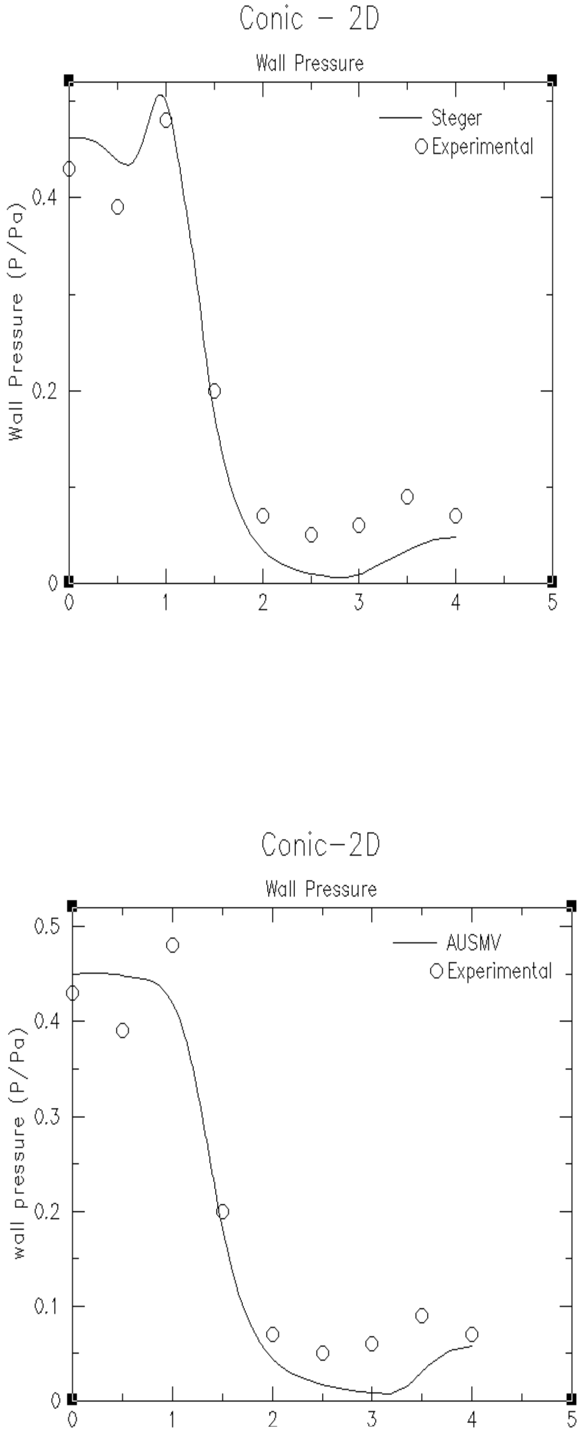


Figure 115

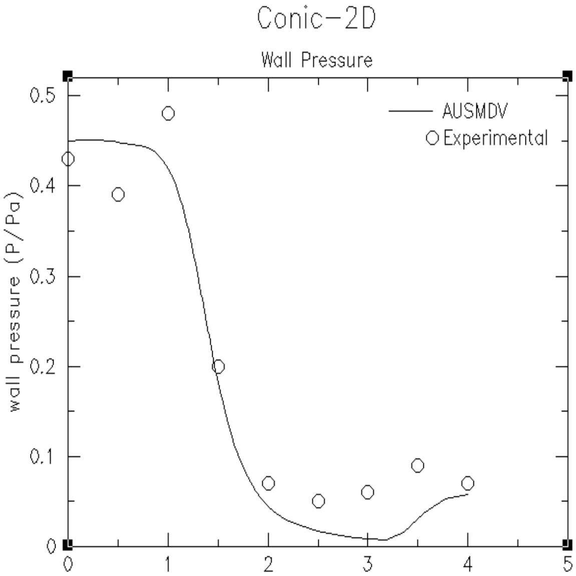


Figure 116

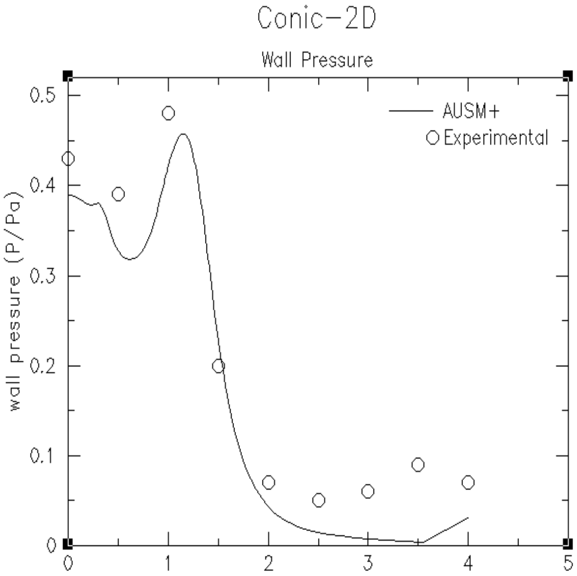


Figure 117

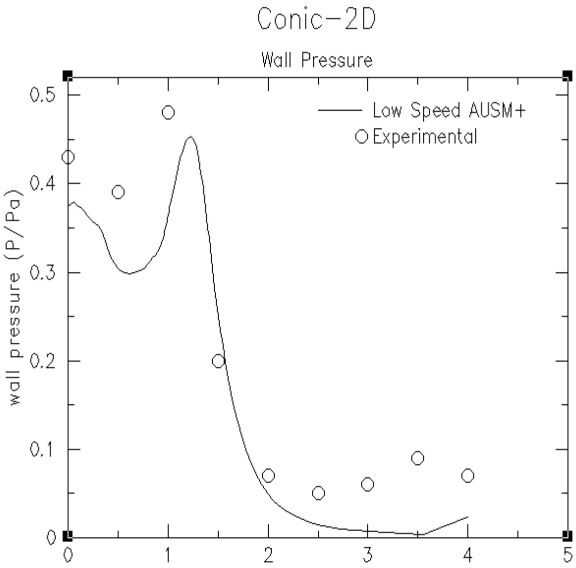
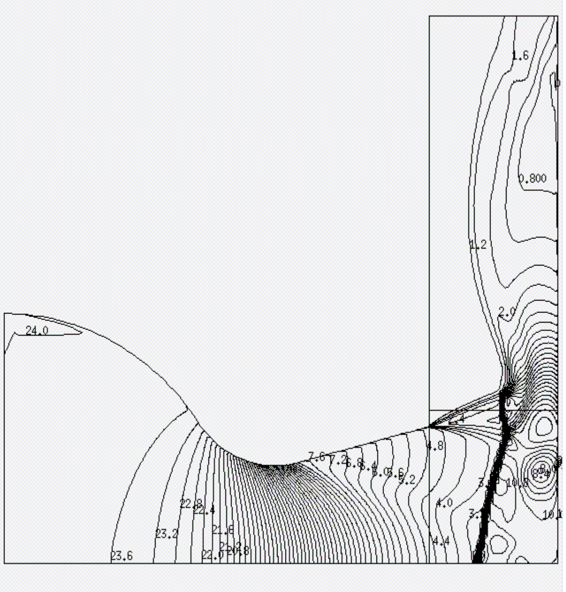
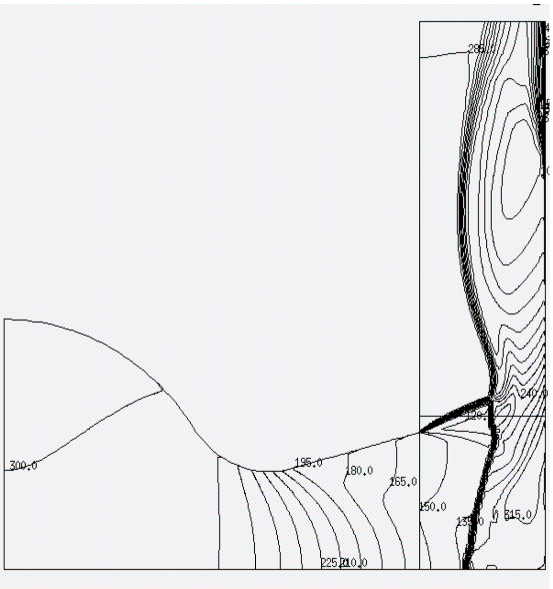


Figure 118

Roe

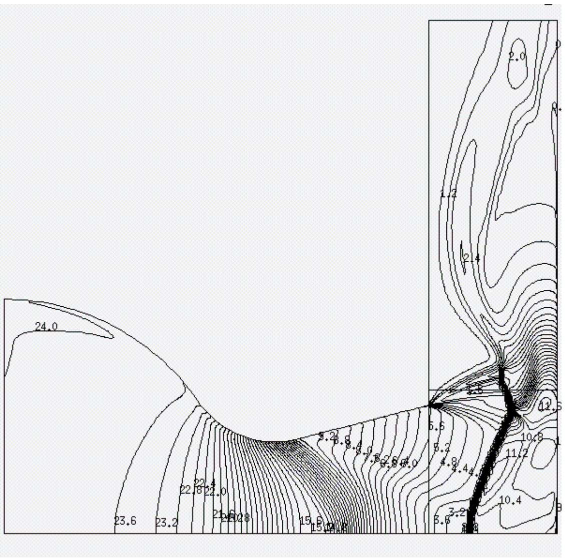


Density
(kg/m^3)
Figure 119

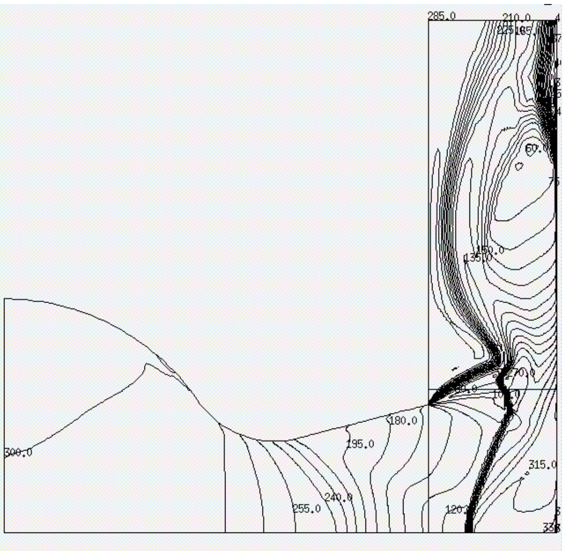


Temperature
(K)
Figure 120

Steger

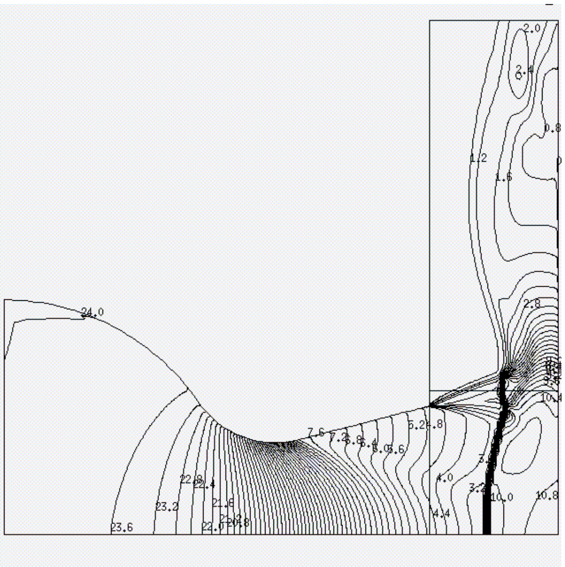


Density
(kg/m^3)
Figure 121

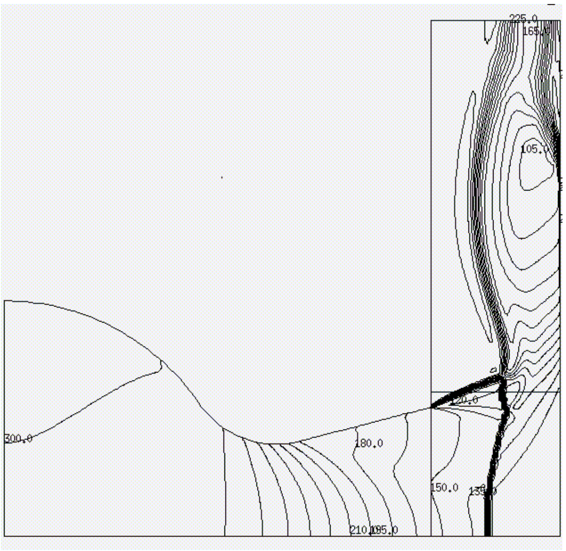


Temperature
(K)
Figure 122

Van Leer

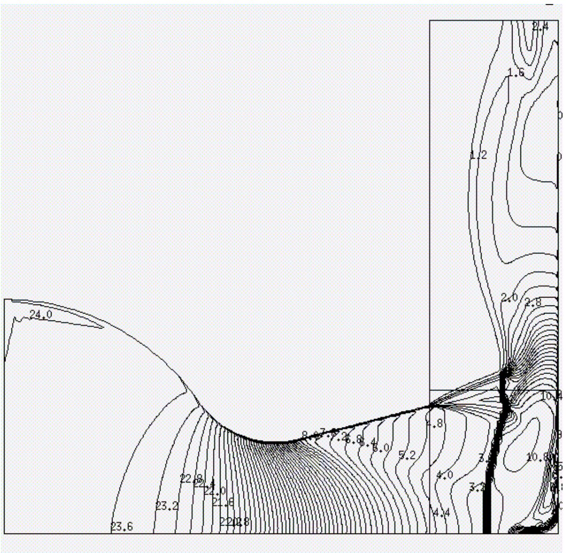


Density
(kg/m^3)
Figure 123

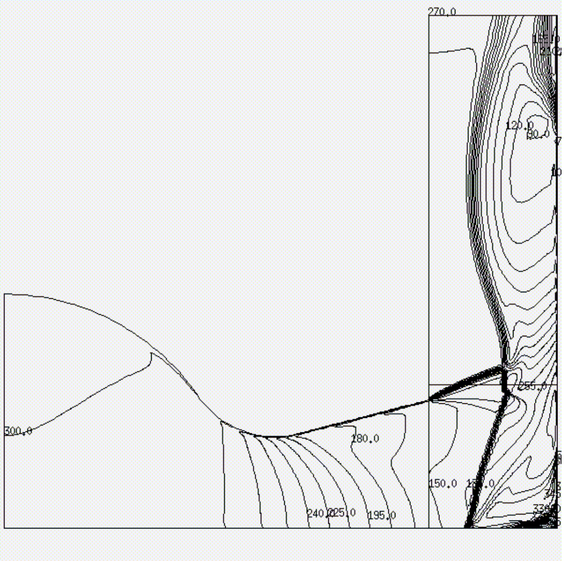


Temperature
(K)
Figure 124

AUSMDV

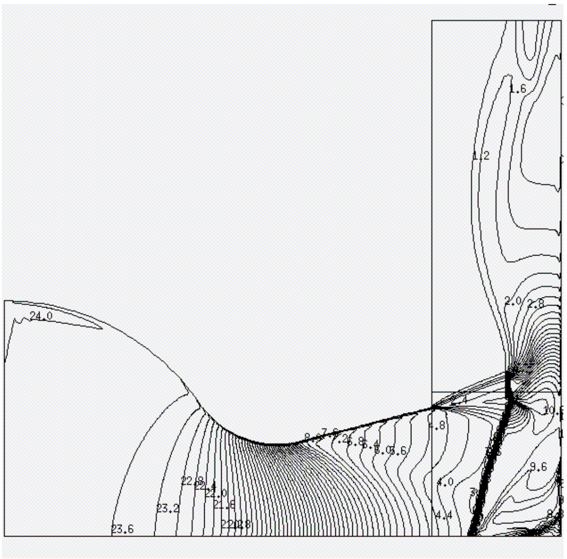


Density
(kg/m^3)
Figure 125

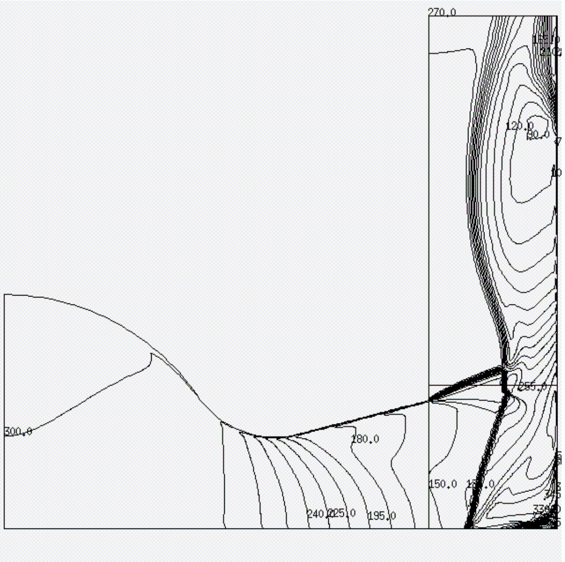


Temperature
(K)
Figure 126

AUSMV

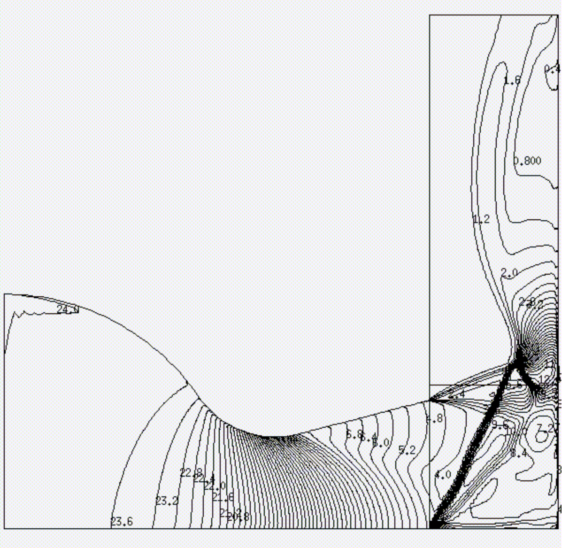


Density
(kg/m^3)
Figure 127

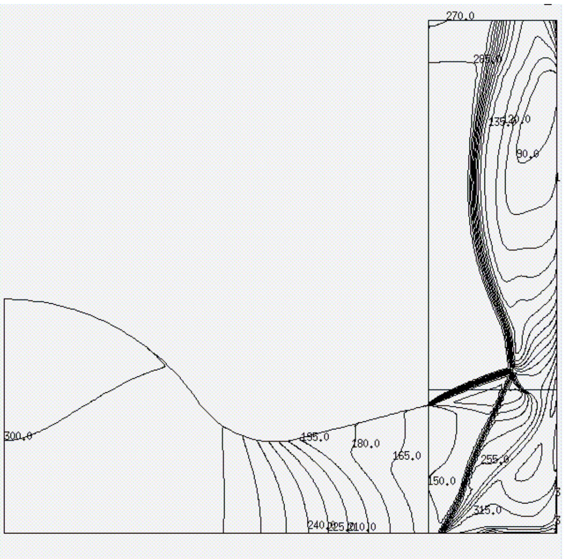


Temperature
(K)
Figure 128

AUSM+

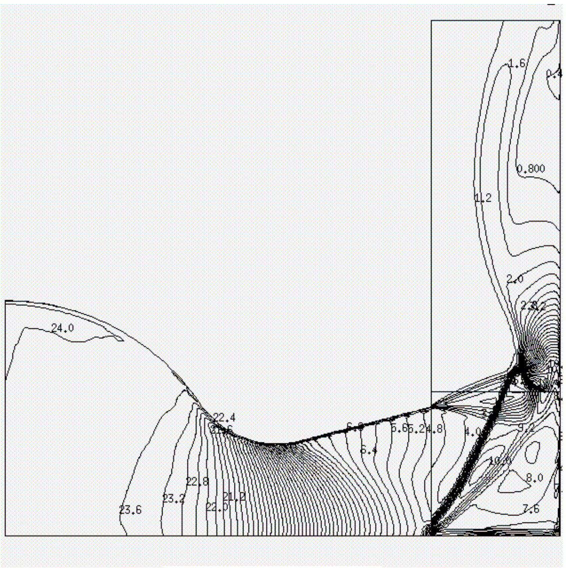


Density
(kg/m^3)
Figure 129

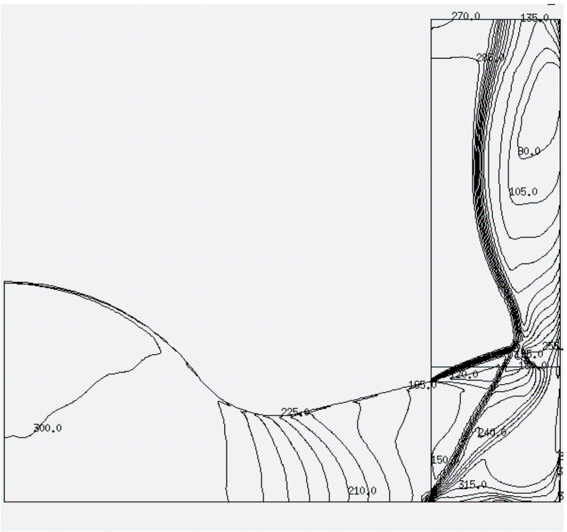


Temperature
(K)
Figure 130

Low Speed AUSM+

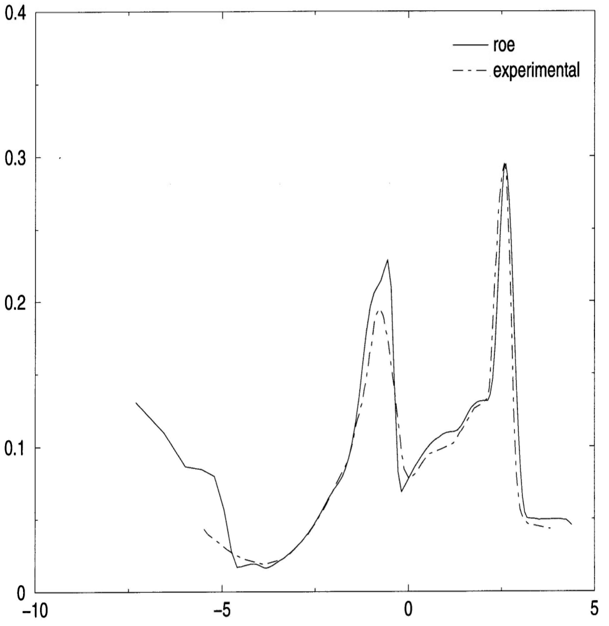


Density
(kg/m^3)
Figure 131



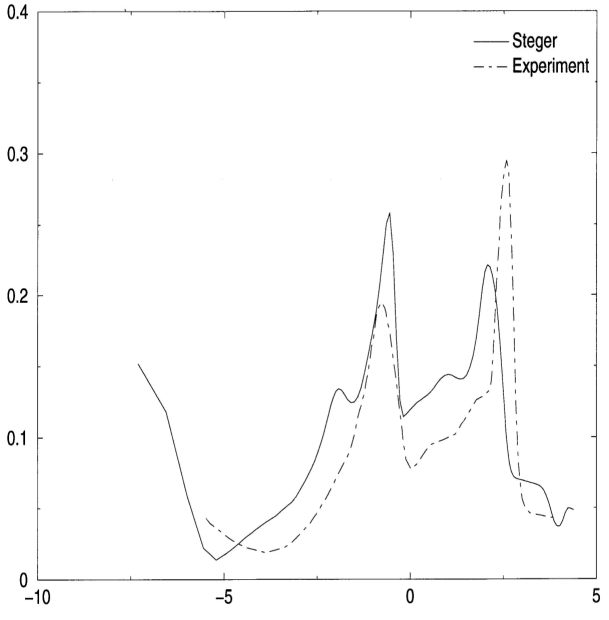
Temperature
(K)
Figure 132

Comparison of Plate Pressure Distribution



S/R$_N$

Figure 133



S/R$_N$

Figure 134

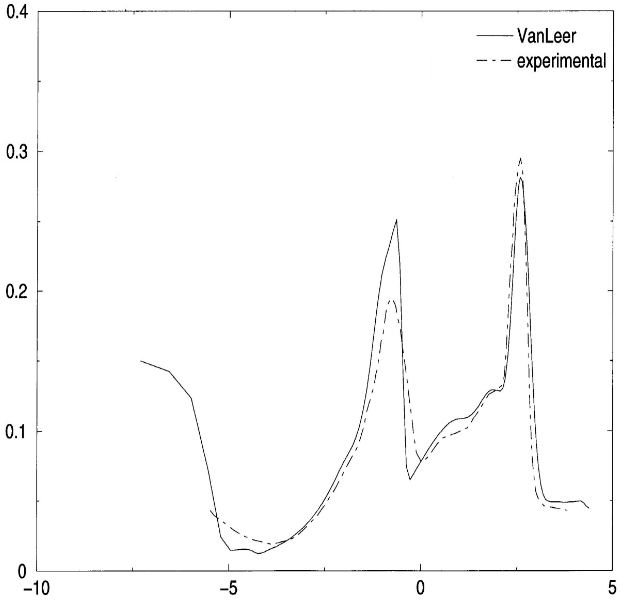


Figure 135

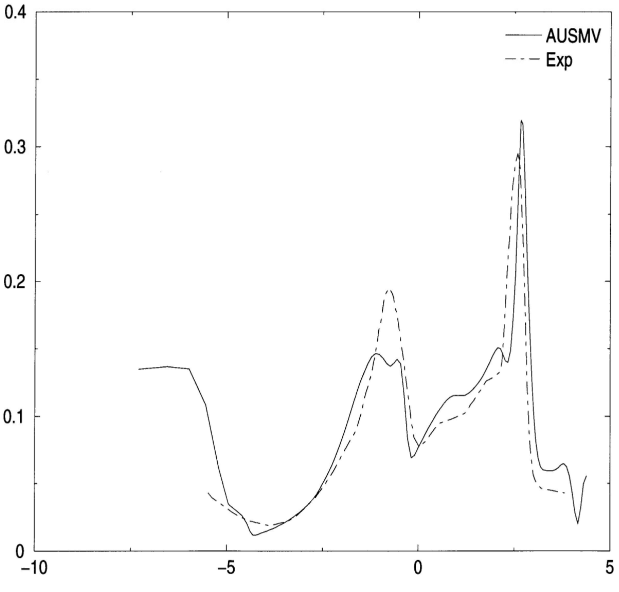


Figure 136

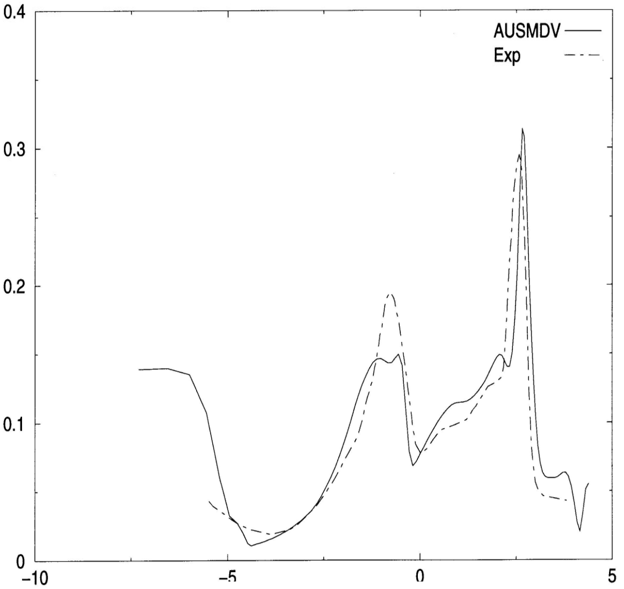


Figure 137

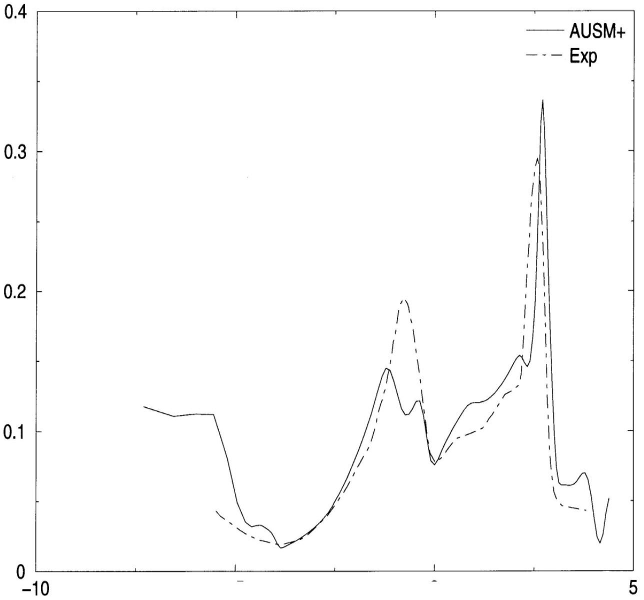
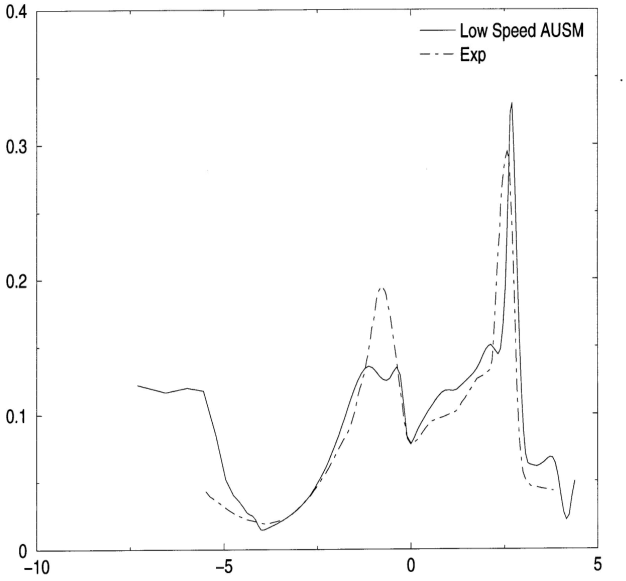


Figure 138



S/R_N

Figure 139

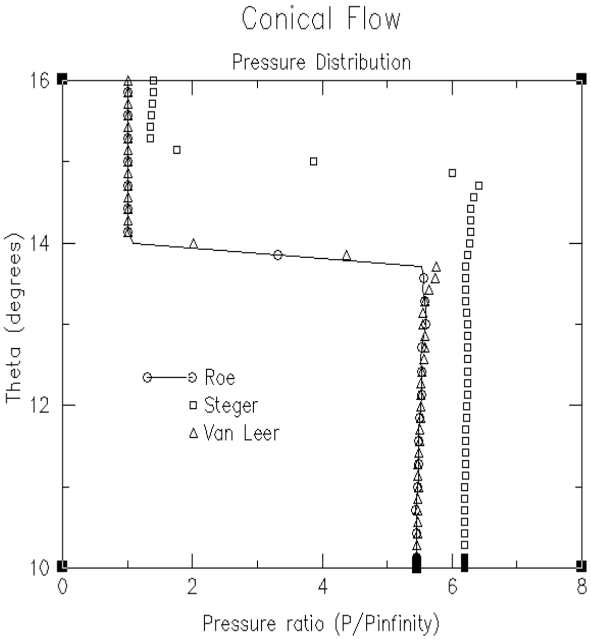


Figure 140

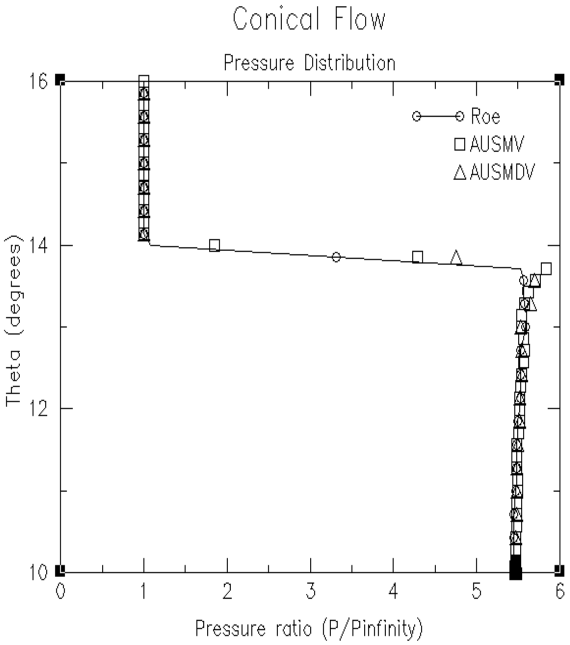


Figure 141

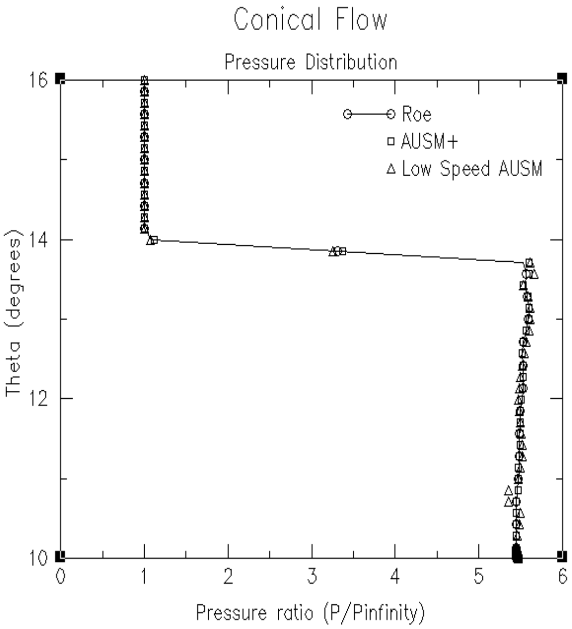


Figure 142

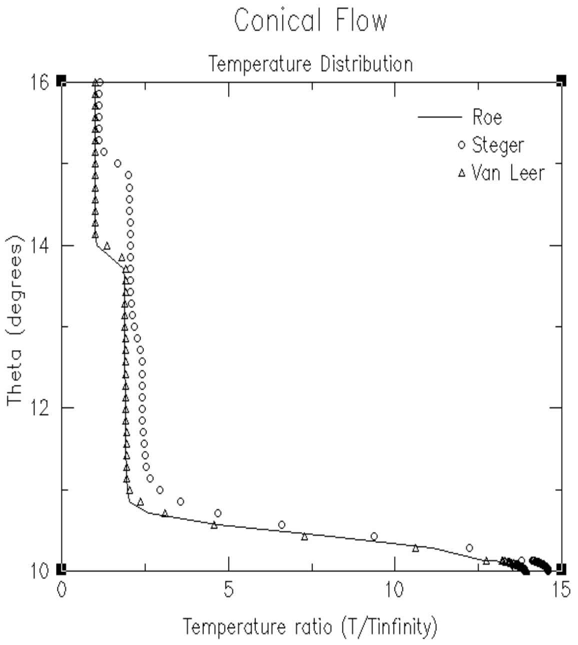


Figure 143

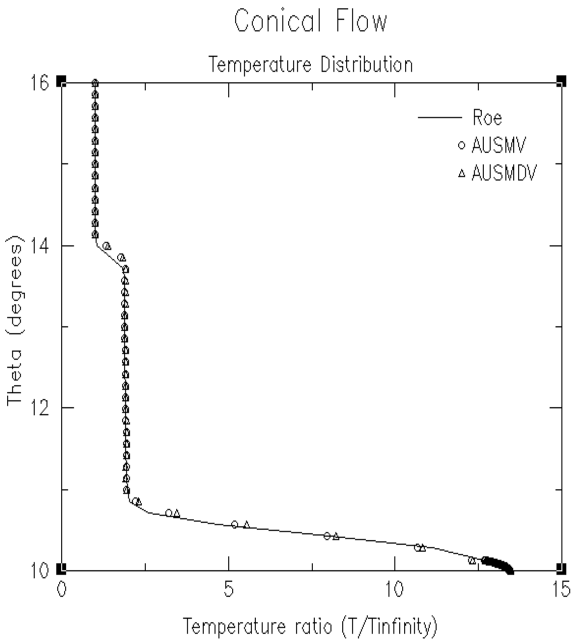


Figure 144

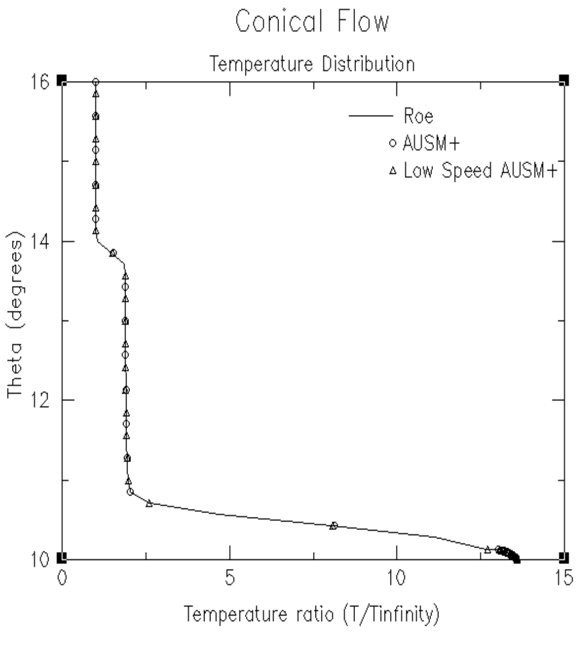


Figure 145

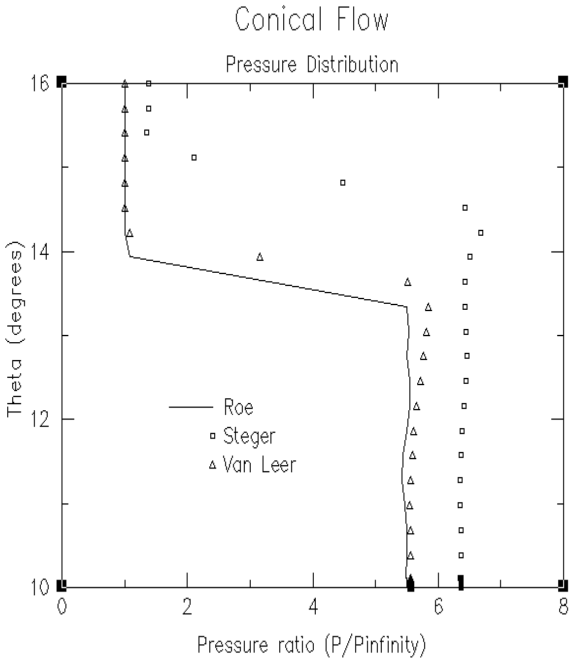


Figure 146

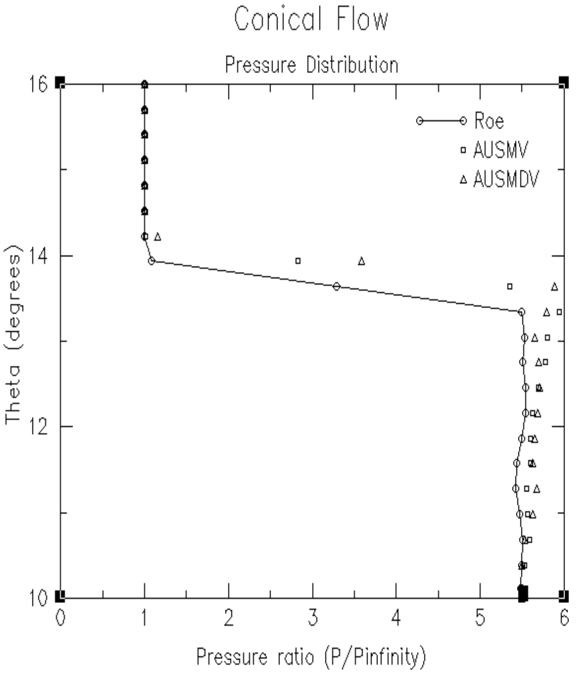


Figure 147

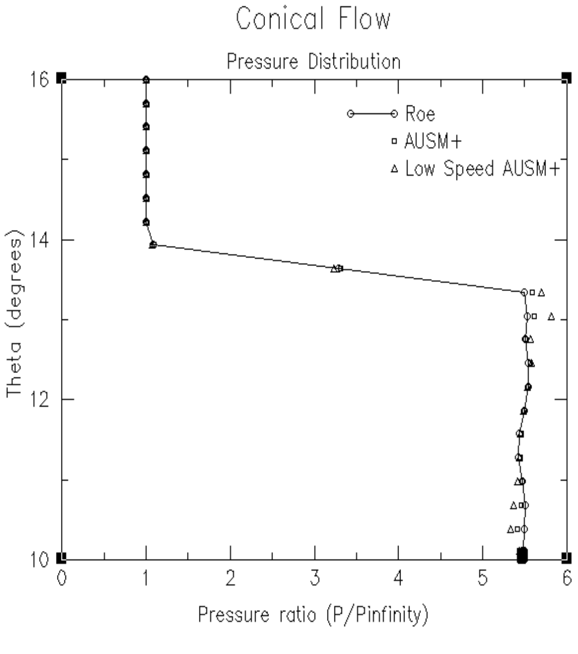


Figure 148

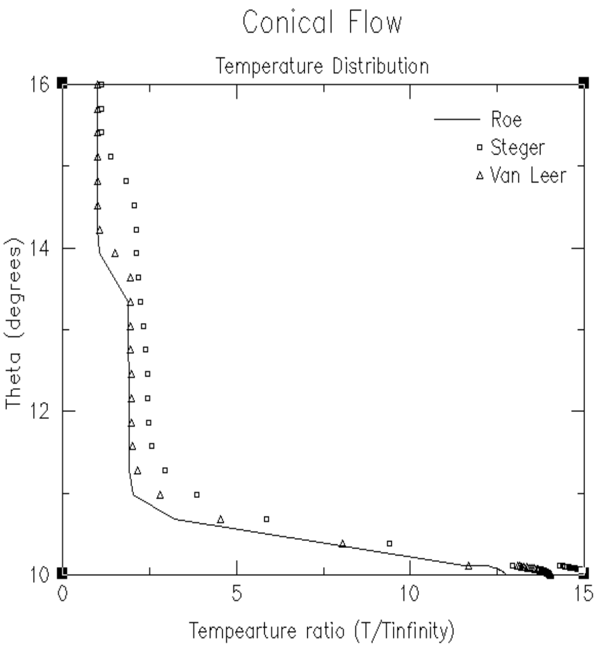


Figure 149

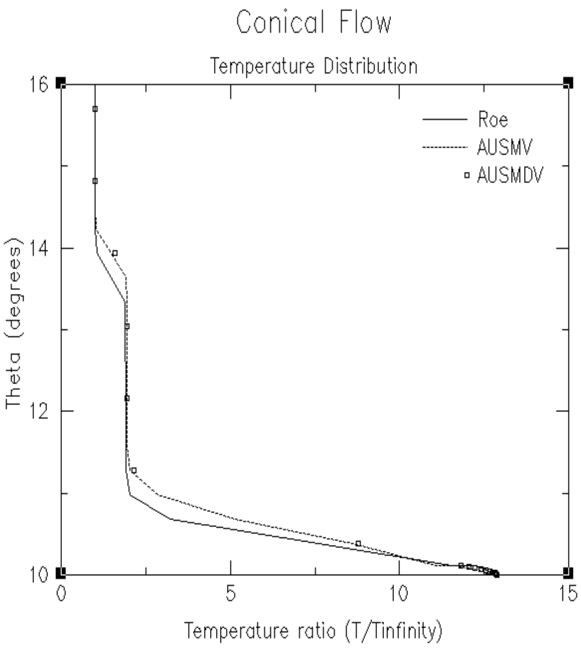


Figure 150

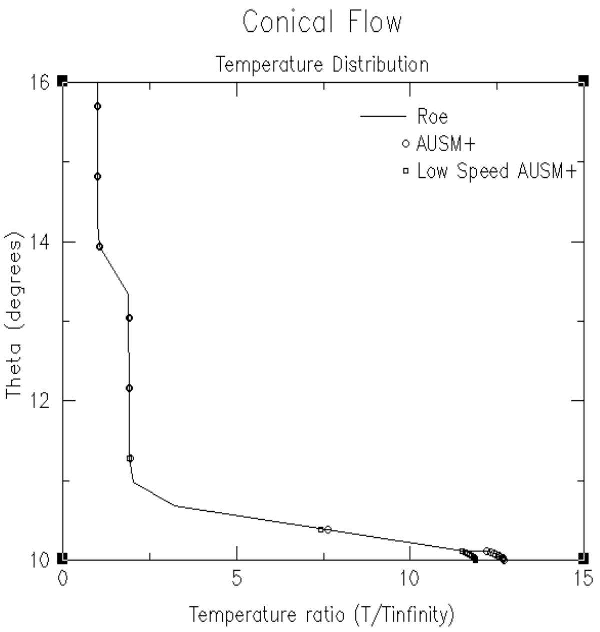
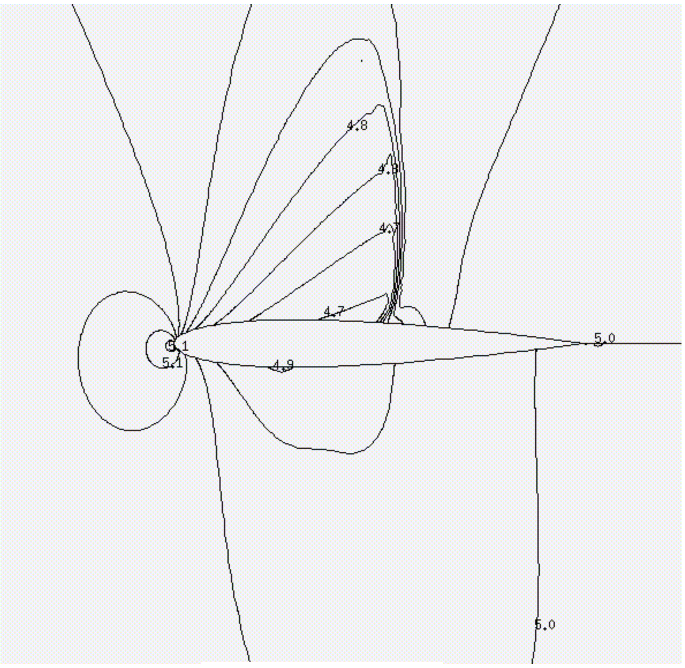
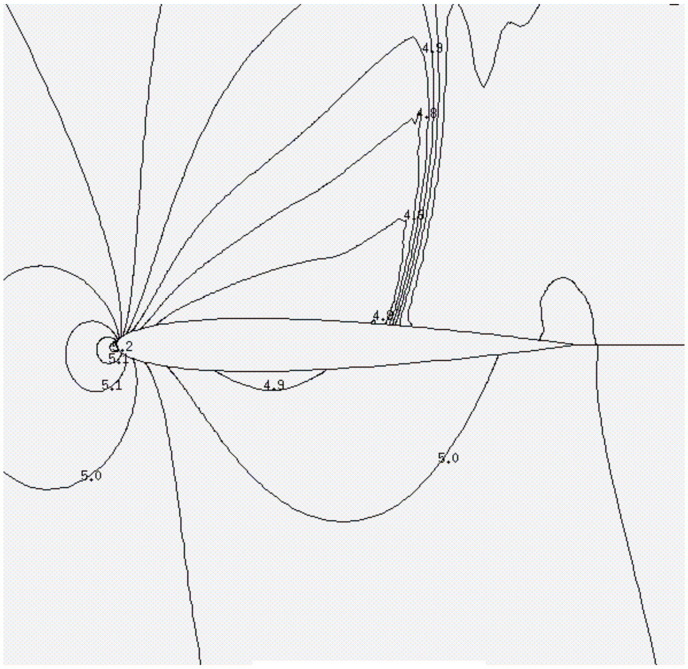


Figure 151

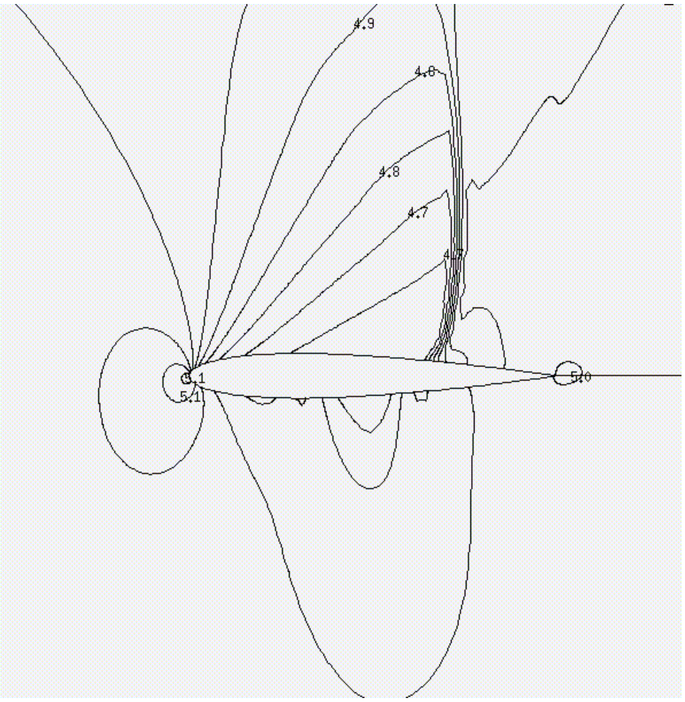
Pressure Contours for an Airfoil
Case A



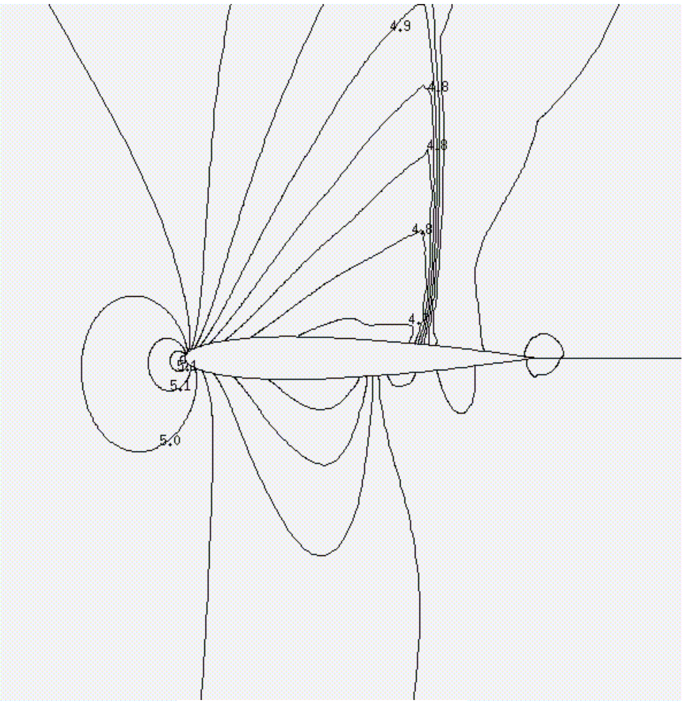
Roe
Figure 152



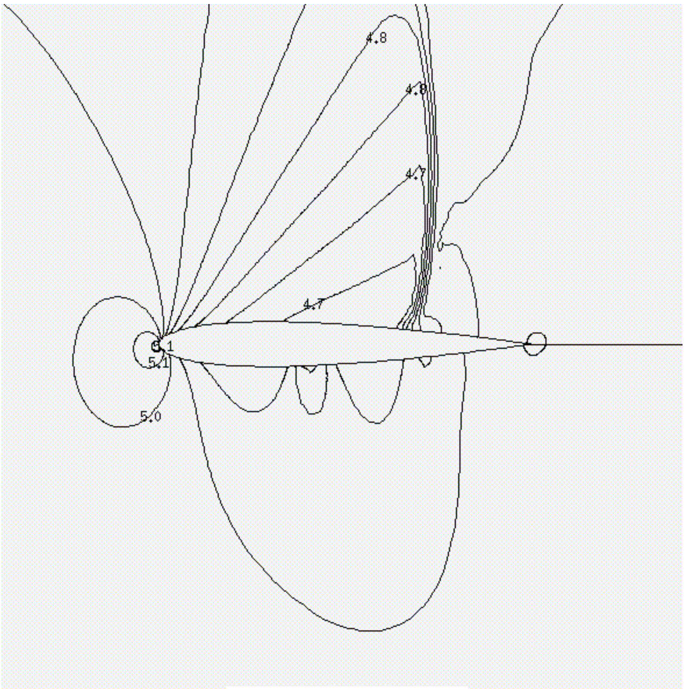
Steger
Figure 153



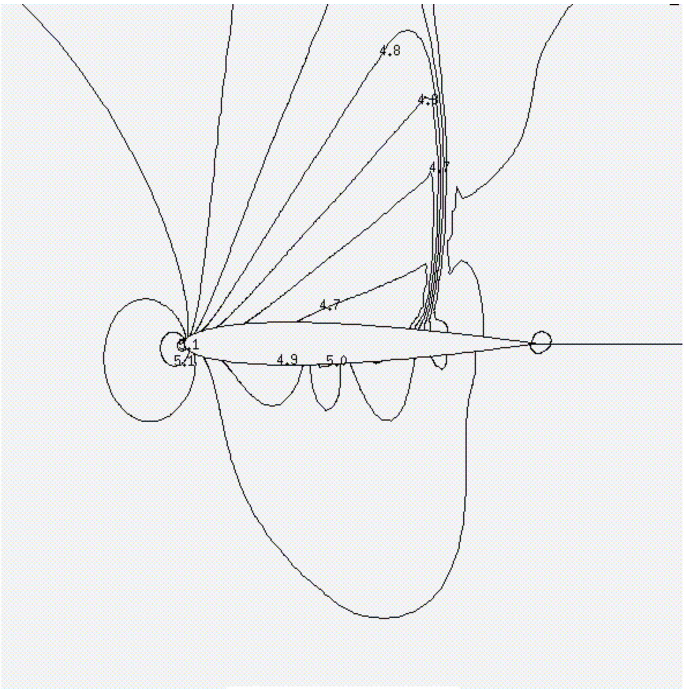
Van Leer
Figure 154



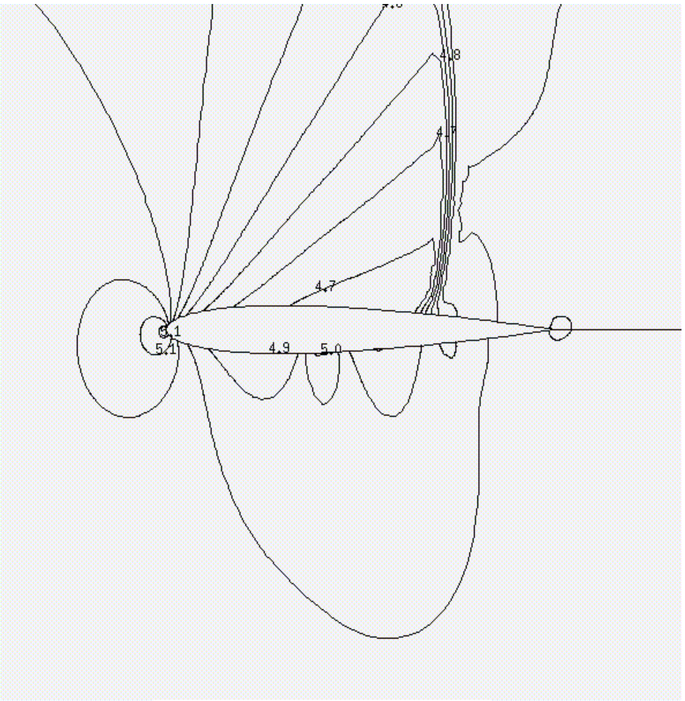
AUSMV
Figure 155



AUSMDV
Figure 156

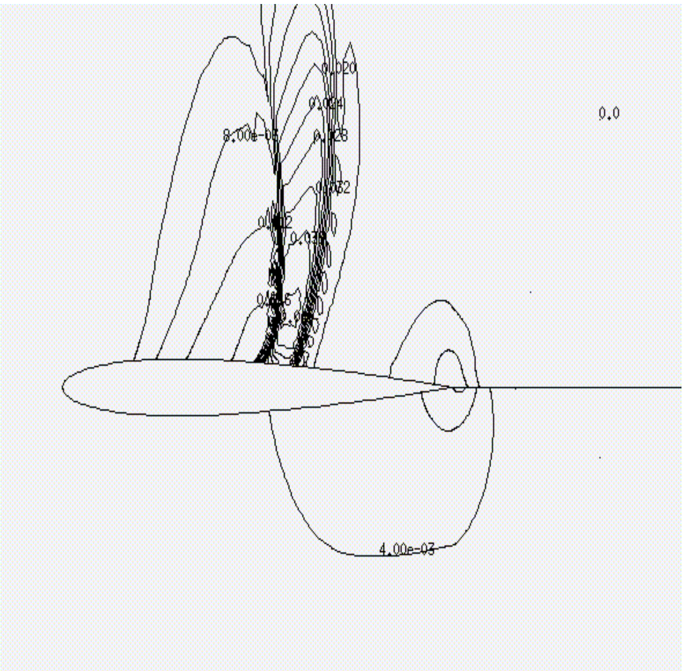


AUSM+
Figure 157

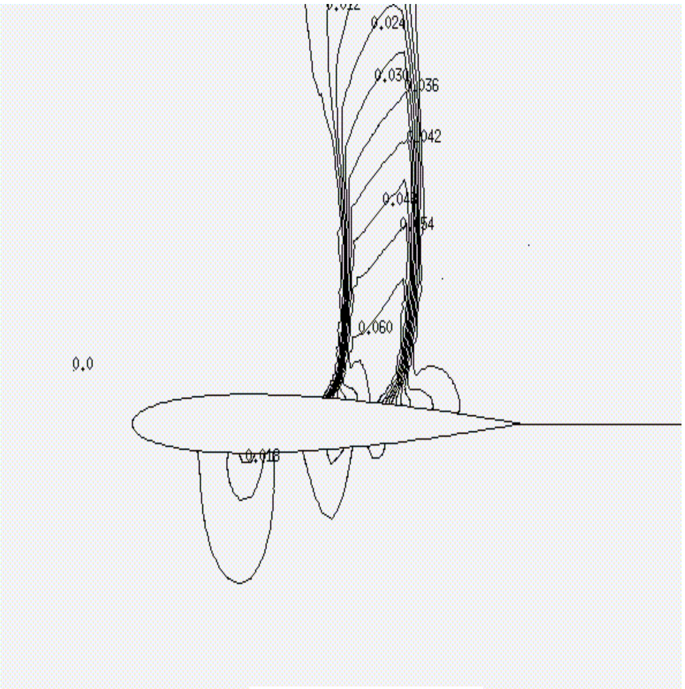


Low Speed AUSM+
Figure 158

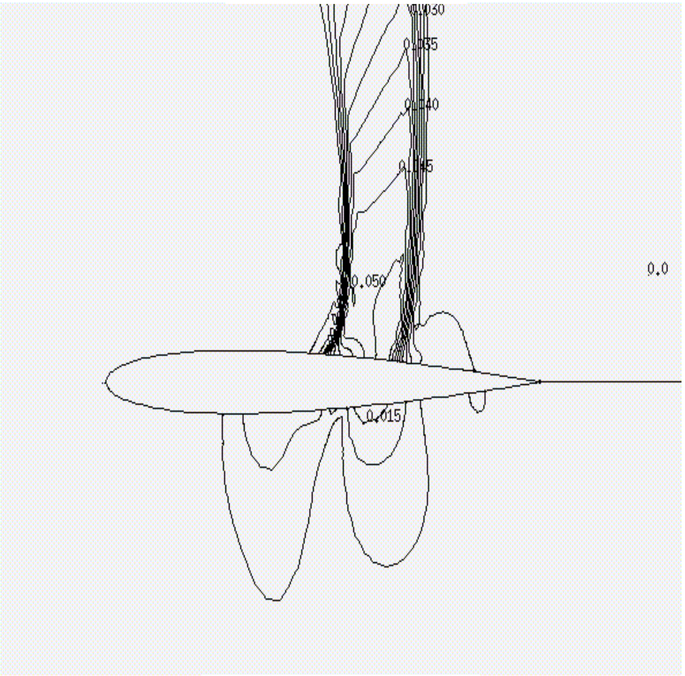
Error Contours for an Airfoil
Case A



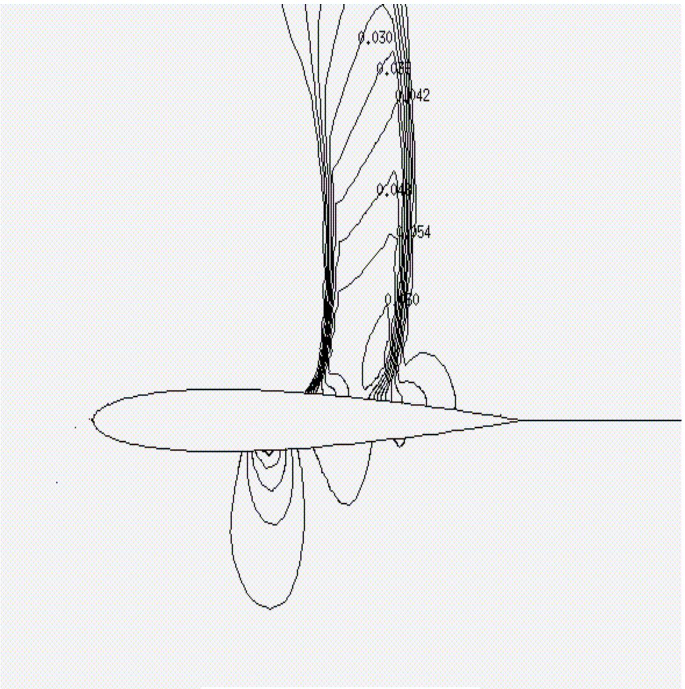
Roe-Steger
Figure 159



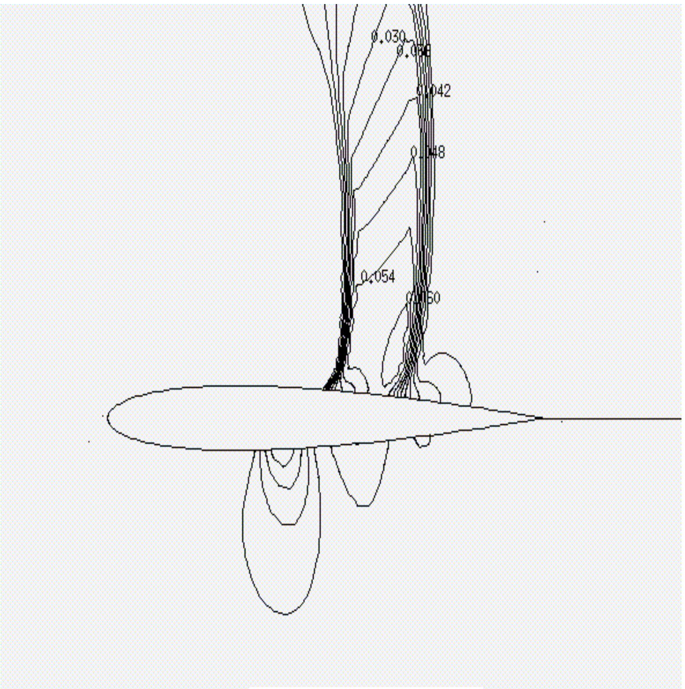
Roe-Van Leer
Figure 160



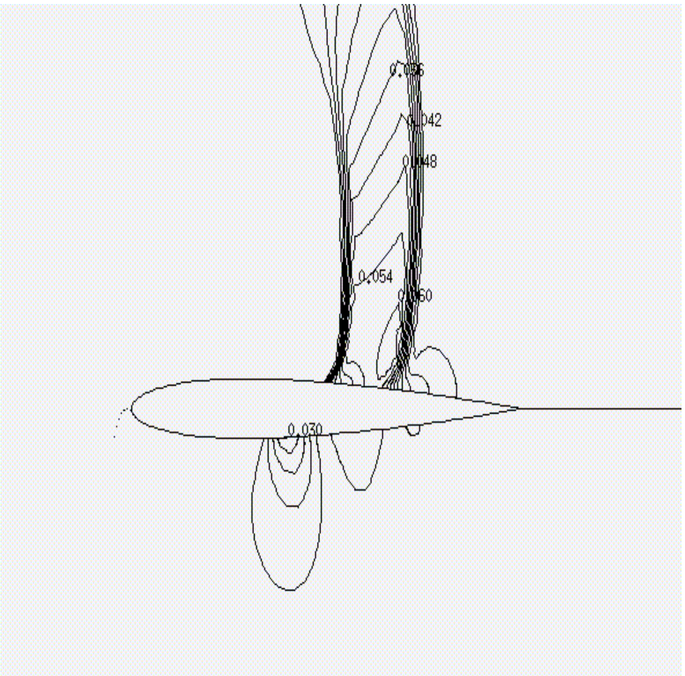
Roe-AUSMV
Figure 161



Roe-AUSMDV
Figure 162

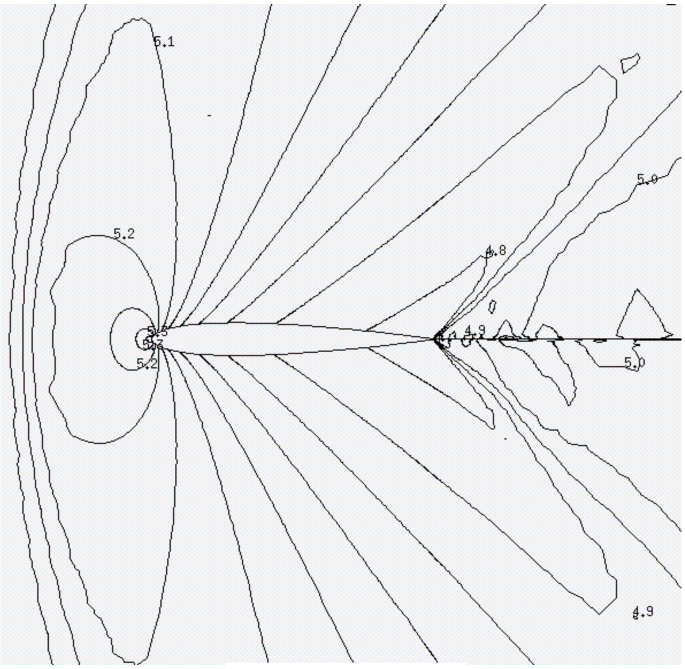


Roe-AUSM+
Figure 163

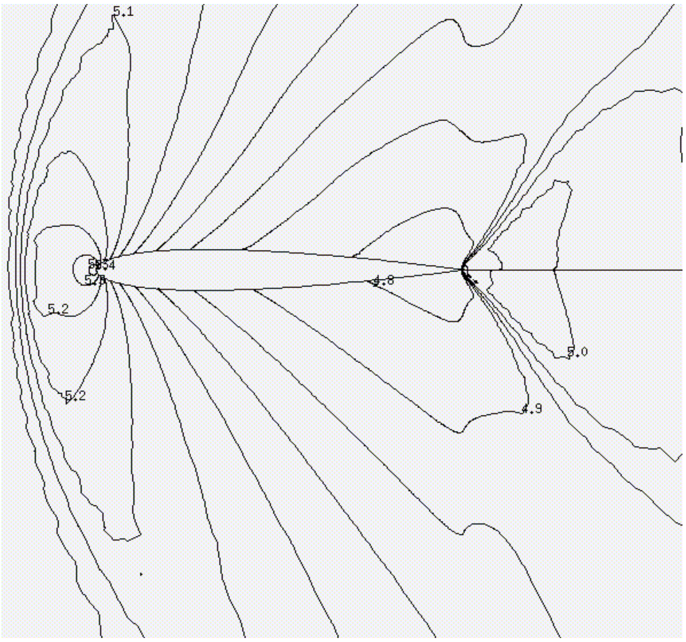


Roe-Low Speed AUSM+
Figure 164

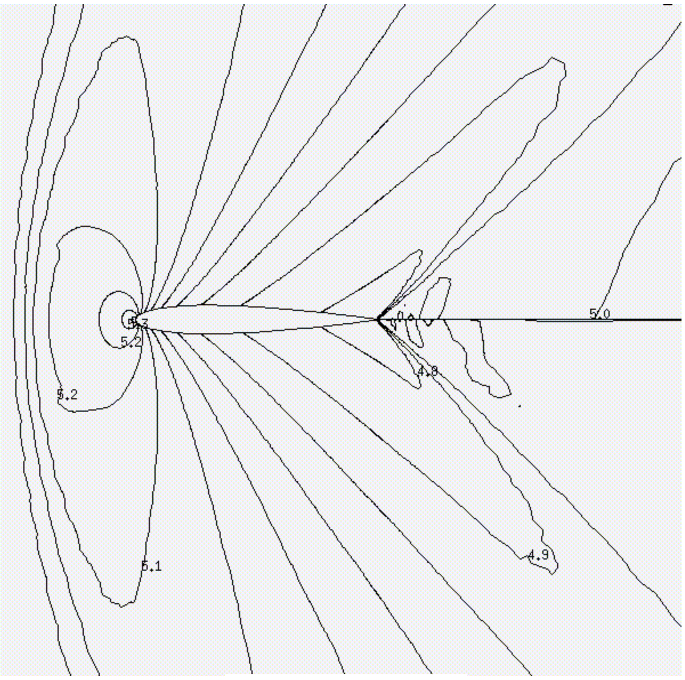
Pressure Contours for an Airfoil
Case B



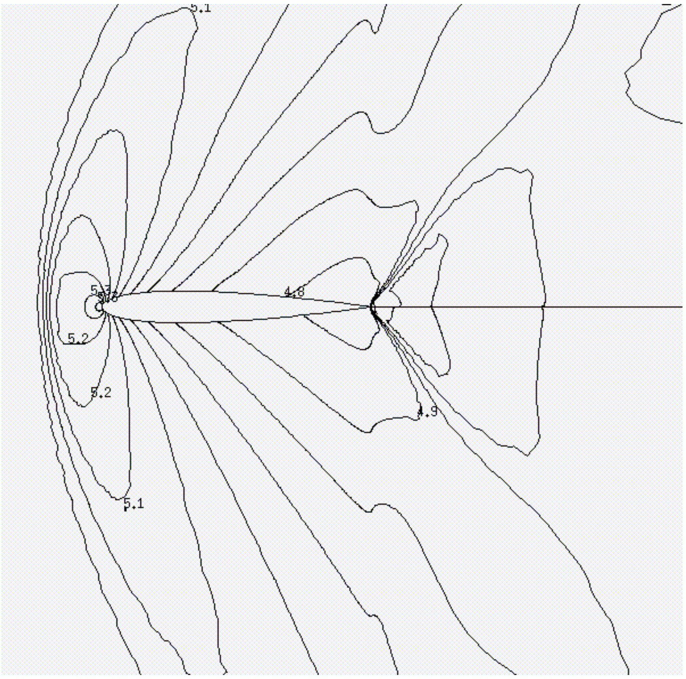
Roe
Figure 165



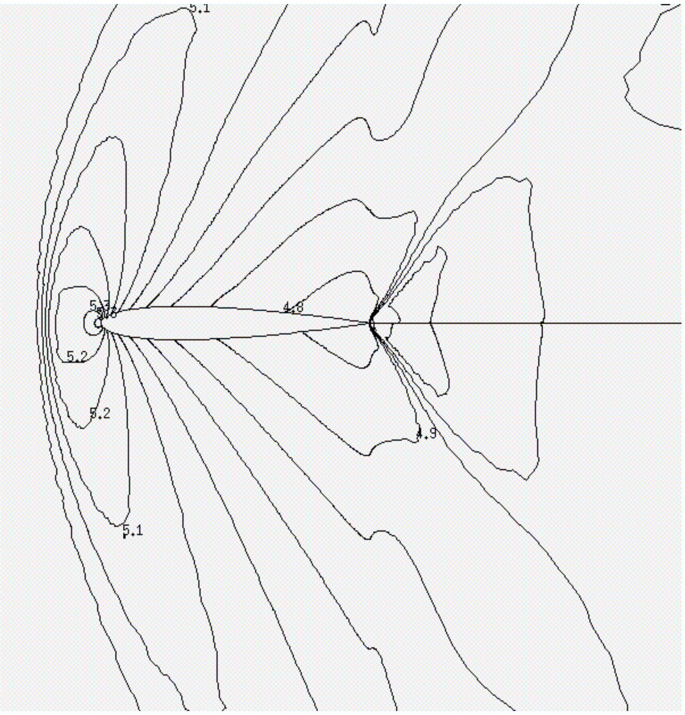
Steger
Figure 166



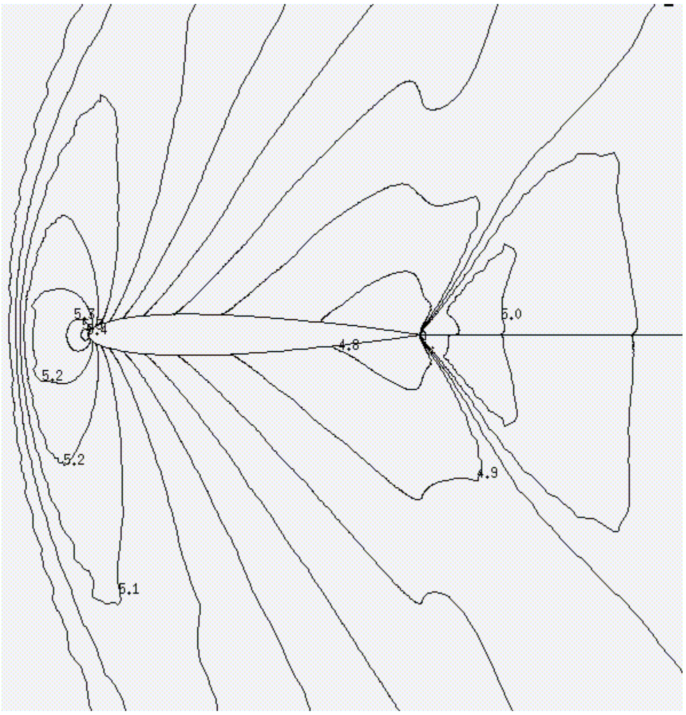
Van Leer
Figure 167



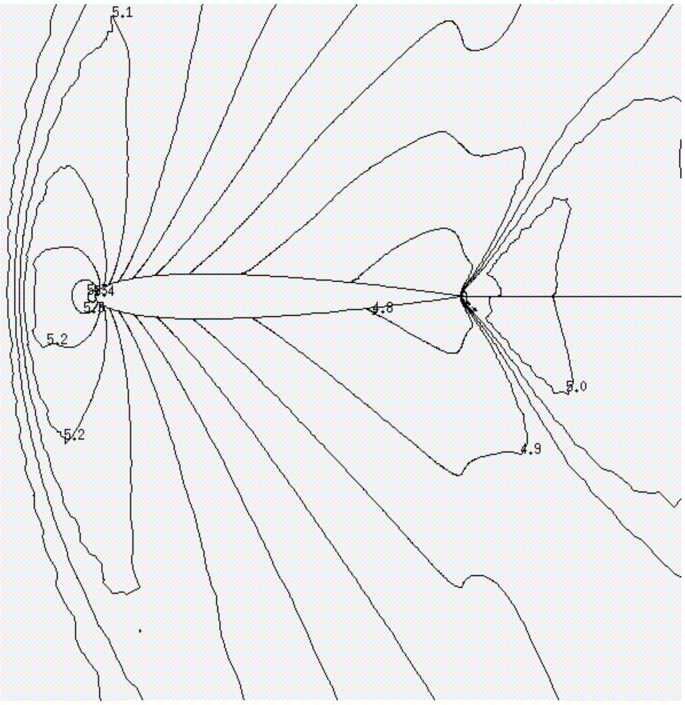
AUSMV
Figure 168



AUSMDV
Figure 169

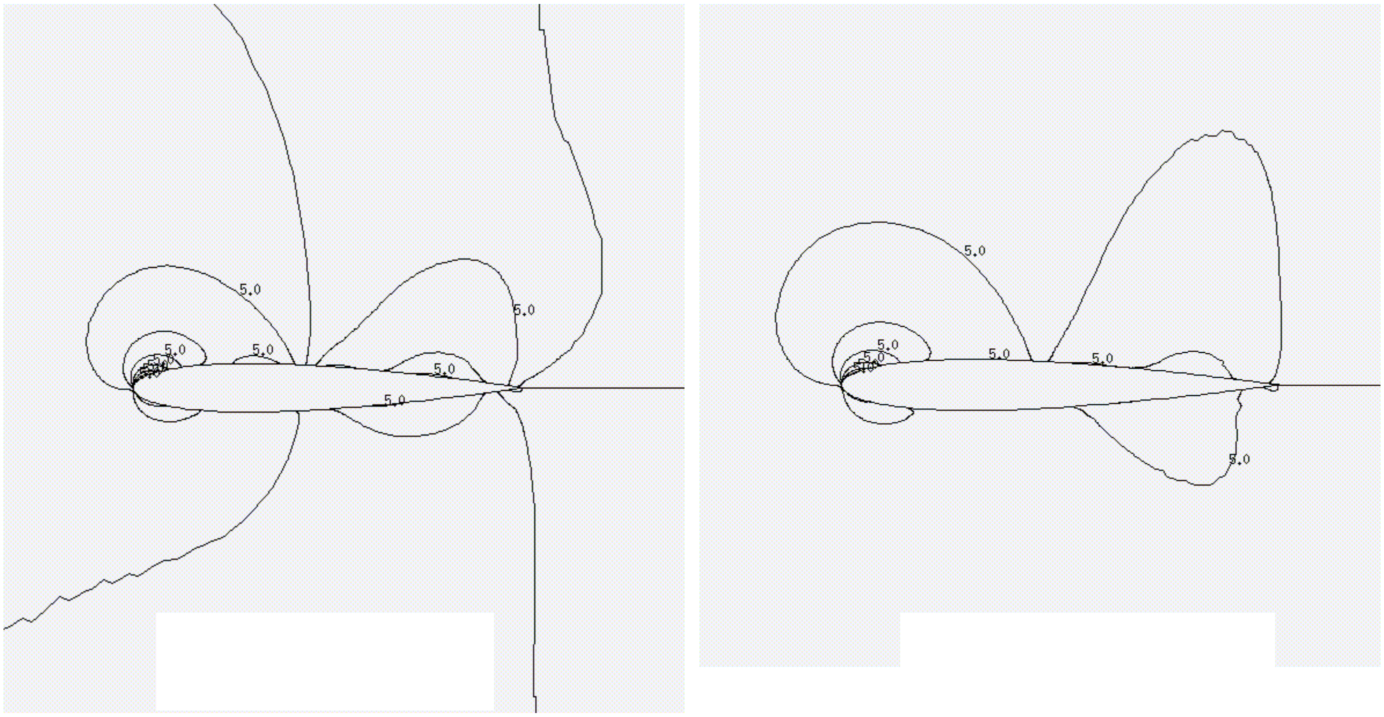


AUSM+
Figure 170

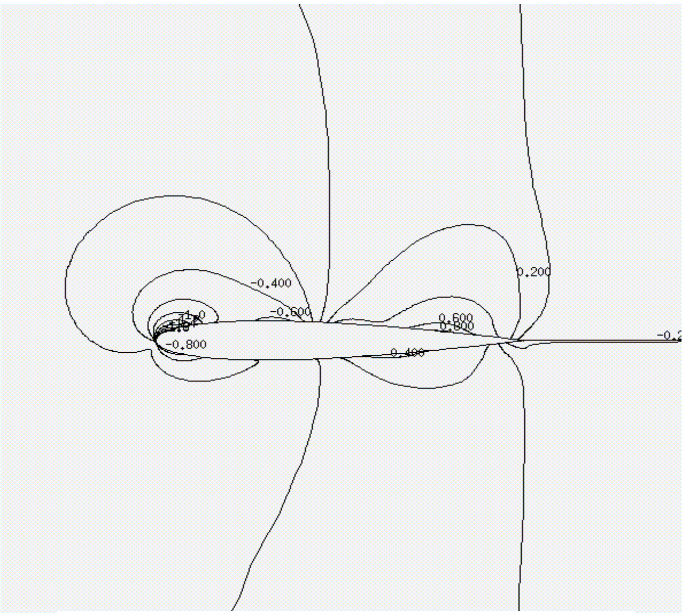


Low-Speed AUSM+
Figure 171

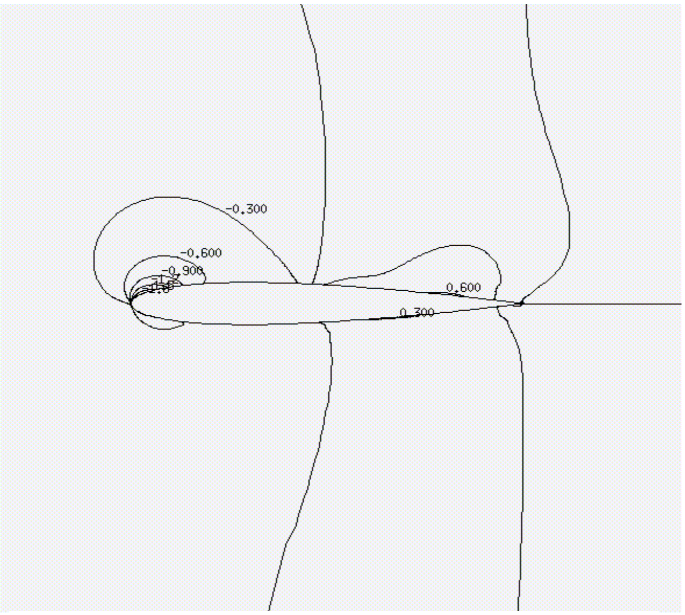
Pressure Contours for an Airfoil
Case C



Pressure Coefficient Contours for an Airfoil

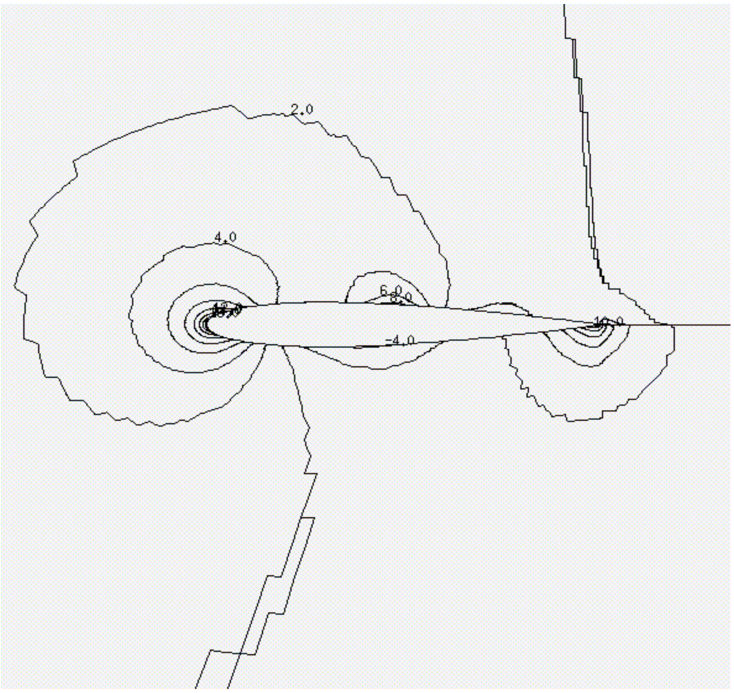


Roe
Figure 174

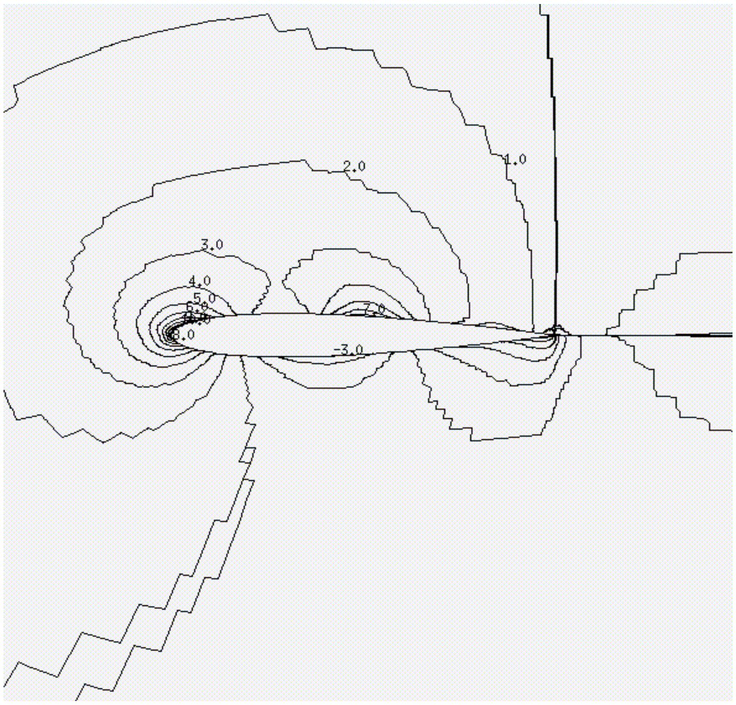


Low Speed AUSM+
Figure 175

Pressure Coefficient Contours for an Airfoil
Case D



Roe
Figure 176



Low Speed AUSM+
Figure 177