Memoirs of the School of Engineering, Nagoya University
Vol.46, No.1 (1994)

ROBUST EULER CODES FOR HYPERSONIC REACTIVE FLOWS

Michel H. LEFEBVRE and Toshi FUJIWARA

Department of Aerospace Engineering

(Received May 31, 1994)

Abstract

The present work describes Codes XXX1 and XXX2, a two-dimensional model for shock-induced combustion. Three-dimensional effects can also be investigated using axisymmetric geometry. The code was initially developed as research tool to model the supersonic combustion induced by a hemispherical body. It has been generalized to model various cases of supersonic combustion. The potential simulations range from reactive flowfield around hypersonic bodies to RAMAC-device geometries. The physical model deals with the Euler conservation equations, using an ideal but calorically imperfect gas law. The heat release mechanism integrates a detailed set of chemical reaction rates. The numerical analysis is based on an explicit TVD scheme for the conservation equations and an implicit integration of the chemical reaction processes. The numerical implementation of the supersonic reactive flow model is described in detail, and typical applications are provided for illustration. These sample calculations show the versatility of the code to achieve high degree of convergence in steady flow problems and to investigate unsteady flowfields.

CONTENTS

1. Introduction

Shock-induced combustion, ranging from decoupled shock-deflagration phenomena to overdriven detonation waves, has been experimentally studied for many years. The possibility of using shock-induced combustion in various propulsion devices has been recently a subject of many investigations[1,2]. Supersonic combustion is considered as a promising potential

combustion-mode in oblique detonation wave engine (ODWE) and supersonic combustion ramjet (SCRamjet). More recently a the University of Washington, an experimental and theoretical research is being carried out on the feasibility of accelerating projectiles to high velocities using a ramjet-in-tube technique[3,4]. The concept is called 'ram-accelerator' (RAMAC). The ram accelerator is an in-bore ramjet device in which a projectile shaped like the centerbody of a supersonic ramjet is propelled down in a stationary tube filled with a combustible gas mixture. Combustion on and behind the projectile generates thrust which accelerates it to high velocities. The acceleration can be tailored using different reacting gas mixtures. RAMAC is a promising technique for launching heavy, acceleration-insensitive payloads into space and offer unique opportunities for a large-scale hypersonic ground test facility.

There are different modes of propulsion for the ram accelerator. They differ primarily by the method of the heat release and the velocity of operation[3,5]. They include subsonic combustion modes and normal and oblique detonation modes.

Numerical simulation of the ram-accelarator has been an ongoing research in many laboratories[6,7]. The present work develops a numerical model in order to simulate flow and combustion processes around a wedge or axisymmetric body in general and in a RAMAC-device in particular. The challenge when simulating such phenomena is to model properly the whole range of combustion processes. Moreover, the difficulty associated with such computation is the integration of both the fluid dynamics conservation equations and the set of chemical rate equations. Most of the differential equations describing the kinetics of detonative mixtures are stiff and the time scales associated with the chemistry are usually much smaller than the time scale of fluid motion.

Therefore, two approaches for solving both types of equation are currently used[8-10]. One way is to uncouple the two types of equations. Each time step consists of an overall gasdynamics step with frozen chemistry, followed by a chemical reaction step of several small time steps at constant total density. The chemistry is integrated into the global timestep before it is coupled to the other processes. The main advantage of this method is that chemistry can be integrated explicitly, which makes the integration algorithm much simpler. This method works well in a practical sense when the changes in the physical quantities (specifically the energy release) during the global timestep are small enough. The major drawback of the explicit timestep-splitting approach is the requirement of subcycling. The timestep limit imposed by some chemical reactions may be orders of magnitude lower than that required by the other physical processes, and so the chemistry can be hunderds of times subcycled before integration over the global timestep.

A second way is to solve the full set of equations simultaneously. This approach requires an implicit treatment for the equations describing the kinetic mechanism. Recently, algorithms have been successfully developed for computing reacting flows based on this approach[11,12].

In the present work, we use an upwind total variation diminishing algorithm (Upwind–TVD), the convection and chemical sources terms being solved explicitly and implicitly, respectively. This method is known as semi-implicit scheme.

2. Overview

The aim of this work is to produce a tool for further study and analysis of shock-induced combustion. The goal is to make possible investigation of various geometry parameters and different combustion regimes. The codes are specifically designed to solve the Euler conservation equations, including chemical source terms, in a general coordinate frame.

Codes XXX1 and XXX2 represent a set of codes that can be used for computing high velocity nonreacting and reacting flows, respectively. Although the codes are able to handle different equivalent ratios and dilutions, the reactive mixture has to be hydrogen–oxygen. The diluent is nitrogen. Further implementation of other diluents can be readily performed by implementing additional thermodynamical data. Reactive mixtures containing fuel other than hydrogen would require implementation of a complete new set of chemical rates. The H_2-O_2 kinetic mechanism used is specifically adapted for hypervelocity flow and has been severely tested and compared with experimental data[13,14].

The overall geometry and boundary conditions are given in Fig. 2.1. By changing geometry-data, the actual geometry can range from a conical blunt body to a conical head smoothly connected to a cylindrical or conoid afterbody. Change in boundary conditions enables the user to simulate projectiles in a round tube, similar to a RAMAC device, or flying bodies.

The numerical method used is an explicit upwind TVD scheme, second-order in time and in space. The chemistry part is solved by implicit finite differences and is implemented into the general scheme by time-splitting technique. The overall scheme presents thus the characteristics of a semi-implicit method and ensures good convergence of the computation. Although the initial purpose of the codes is to model reactive flowfields, simulation of hypersonic nonreacting flow can be usefully performed.

The core of the algorithm was developed by Matsuo[14] and has been successfully used to model combustion instabilities around hemispherical supersonic bodies. Results of these computations perfectly agreed with previous experimental works[13,15], proving the ability of the code to simulate satisfactorily supersonic reactive flow.

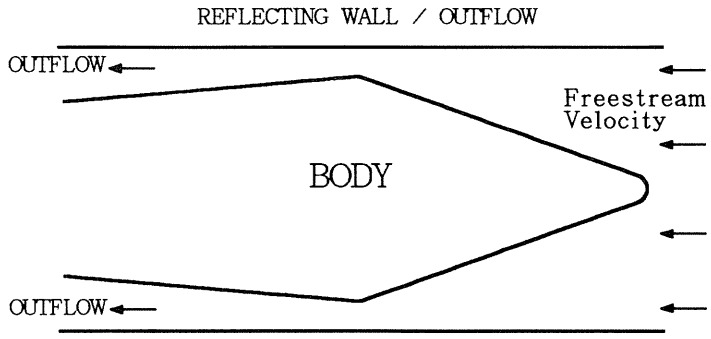


Fig. 2.1. Overall geometry and boundary conditions implemented in the present code.

After reviewing the physical and chemical equations used in the present work (section 3), section 4 deals with the general numerical scheme and particular numerical techniques used to perform the computation. The next section describes both the nonreactive and reactive codes and should enable any inexperienced user to readily start using the codes. Follows a series of sample calculations performed with both nonreactive and reactive systems. These are part of sections 6 and 7, respectively. The purpose of these sample computations is to demonstrate the versatility of the codes and to supply the potential user with some basic results. Some of the performed calculations exhibit more profound physical implications and are then discussed in more details.

3. Physical Analysis

The present section discusses the basic equations describing supersonic reactive flows. The fundamental equations of the fluid dynamics are simplified to form the physical model used in the numerical code. A separate item deals with the chemical model which provides appropriate simulation for heat release.

3.1. Governing Equations

Numerical simulation of supersonic flow is based on the compressible, time-dependent, conservation equations for total mass density ρ, momentum $\rho\hat{v}$, energy E and individual species mass densities ρ_i. Depending on the complexity and the nature of the physical phenomena involved, these equations include none or some of the following physical processes: heat transfer (radiation, thermal convection and conduction), mass diffusion, viscosity, chemical energy. Depending on the system studied, various forces can be taken into account, like gravity and electromagnetic forces. Because we are dealing here with supersonic reactive systems which are characterized by their high velocity, most of the aforementioned phenomena can be neglected and the set of conservation equations is simplified to become the inviscid Euler-equations. Simulations of supersonic reactive flow performed with the Navier-Stokes conservation equations have so far shown little influence of the viscosity on the overall shock and detonation structures[16,17].

Assuming a time-invariant grid and using the transformation of coordinates implied by $\xi = \xi(x,y)$ and $\eta = \eta(x,y)$, the cartesian equations can be recast into conservation form where ξ and η are the new independent variables and x_ξ, x_η, y_ξ, and y_η are the four grid metric terms obtained numerically from the mapping procedure

$$x_\xi = \frac{\partial x}{\partial \xi}, \quad x_\eta = \frac{\partial x}{\partial \eta}, \quad y_\xi = \frac{\partial y}{\partial \xi}, \quad y_\eta = \frac{\partial y}{\partial \eta}.$$

For 2D-axisymmetric reacting flow, the equations are

$$\frac{\partial \hat{Q}}{\partial t} = -\frac{\partial \hat{E}}{\partial \xi} - \frac{\partial \hat{F}}{\partial \eta} - \hat{H} + \hat{S}, \tag{1}$$

where $\hat{Q}$ is the vector of basic physical quantities, $\hat{E}$ and $\hat{F}$ are the inviscid flux vectors in the ξ- and η- directions respectively, $\hat{H}$ is the axisymmetric source term vector, and $\hat{S}$ is the chemical reactions source term. They are given by

$$\hat{Q} = J^{-1}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho_i \end{pmatrix}, \quad \hat{E} = J^{-1}\begin{pmatrix} \rho U \\ \rho u U + \xi_x P \\ \rho v U + \xi_y P \\ (E+P)U \\ \rho_i U \end{pmatrix},$$

$$\hat{F} = J^{-1} \begin{pmatrix} \rho V \\ \rho u V + \eta_x P \\ \rho v V + \eta_y P \\ (E+P)V \\ \rho_i V \end{pmatrix} \quad , \quad \hat{H} = J^{-1} \frac{v}{y} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E+P \\ \rho_i \end{pmatrix} ,$$

$$\hat{S} = J^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_i \end{pmatrix} , \tag{2}$$

where terms ω_i are used to keep tracking the transformation among the species resulting from chemical reactions. Their formulation is described in the next section. The inverse of the grid Jacobian J^{-1} is the determinant of

$$\begin{vmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{vmatrix} .$$

The contravariant velocities U and V are defined as follows

$$U = \xi_x u + \xi_y v , \quad V = \eta_x u + \eta_y v ,$$

where coefficients ξ_x, ξ_y, η_x, and η_y are function of the grid metric terms and are given by

$$\xi_x = J y_\eta , \quad \xi_y = -J x_\eta , \quad \eta_x = -J y_\xi , \quad \eta_y = J x_\xi .$$

The equation of state used is the law for thermally ideal gases

$$P(T,\rho) = \sum_{i=1}^{n} \frac{\rho_i}{W_i} \cdot RT = \rho RT .$$

The total density of energy E is the sum of the internal energy and the kinetic energy,

$$E = \sum_{i=1}^{n} \rho_i \mathrm{u}_i + \frac{1}{2}\rho (u^2 + v^2) = \rho \mathrm{U} + \frac{1}{2}\rho (u^2 + v^2) . \tag{3}$$

The internal energy of the system U is related to temperature through the caloric equation of state and consists of internal energy of formation and sensible internal energy, that is,

$$\mathrm{U}(T) = \mathrm{U}_f + \int C_v(T) dT . \tag{4}$$

For a mixture of perfect gases, the total internal energy is given by

$$U(T) = \sum_{i=1}^{n} n_i u_{f,i}^o + \int_{To}^{T} \sum_{i=1}^{n} n_i c_{v,i}(T)\, dT . \tag{5}$$

In this formulation the internal energies of formation $u_{f,i}^o$ are the values at the reference temperature T_o. The temperature is determined from the definition of the total energy (Eq. (3)) and the expression of the internal energy (Eq. (5)). Since the unknown temperature is the upper boundary of the integral in Eq. (5), its computation requires an iterative procedure. It is computer-time saving to use a temperature-independent heat capacity C_v. This approach leads to the use of the heat capacity ratio γ and is commonly used in nonreactive and reactive flow computation. However, it has been shown that modeling chemically reacting flows with the assumption of a constant specific heat can lead to large errors because of overestimation of the temperature[18]. We therefore use in the present code specific heats c_{vi} expressed as a polynomial of temperature.

3.2. *Chemical Kinetics*

The detailed description of the chemical reaction source terms ω_i (Eq. (2)) requires the knowledge of a set of elementary chemical reactions. Terms ω_i calculate the change in density of species i due to the chemical reaction and are given by

$$\omega_i = \frac{d\rho_i}{dt} = W_i \cdot \left[\sum_k \pm \nu_{i,k} \cdot \alpha \cdot \left(k_{k,for} \prod_j C_j^{\xi_{j,k}} - k_{k,back} \prod_j C_j^{\xi''_{j,k}} \right) \right] \tag{6}$$

where the positive or negative sign has to be applied when species i is product or reactant of the forward reaction, respectively. The global third body efficiency α is given by

$$\alpha = \sum_j C_j \cdot \alpha_{j,k}$$

Values of $\alpha_{j,k}$ different from unity are given in Table 3.1. The reverse reaction constant $k_{k,back}$ is related to the forward reaction constant $k_{k,for}$ and to thermodynamic quantities through the chemical equilibrium constant:

$$K_{eq,k}^o = \frac{k_{k,for}}{k_{k,back}} \cdot (RT)^{\Delta\nu_k}$$

$$\ln K_{eq,k}^o = \frac{\pm \nu_{i,k} s_i^o}{R} - \frac{\pm \nu_{i,k} \Delta h_{i,f}^o}{RT}$$

Similarly to Eq. (6), the plus/minus sign refers to species i being product/reactant in the forward reaction, respectively.

Integrating the whole set of differential equations (6) is computer time consuming and many scientists favor the use of parametric models for describing the heat release[19]. Although such calculations have been successfully used to simulate supersonic reactive flow[15,20], we think that good qualitative modeling requires the chemistry to be modeled more

accurately. An increasingly popular option is to use a reduced mechanism which includes three to seven pseudo-elementary chemical reactions. In that case, the drawback is usually the limited range of physical conditions the model can be used for. We decided to use a full set of chemical reactions, that is, a detailed kinetic mechanism for describing terms ω_i. One must be careful when selecting the chemical reactions and rate constants because all mechanisms do not fit well into supersonic combustion data. The set of reactions and rate constants we selected is the one developed by Wilson and MacCormack[21] and modified by Jachimowski[22]. It has been specifically designed for supersonic combustion. However, the mechanism used in the present code does not take into account the species from the reaction of nitrogen. So we use 8 reactive species (H_2, O_2, H, O, OH, H_2O, HO_2 and H_2O_2) and 19 reactions. Table 3.1 shows the mechanism adopted. The initiating reaction considered is

$$H_2 + O_2 \rightarrow HO_2 + H\,.$$

A similar approach has already been successfully used by Matsuo for quantitative modeling of a hypersonic flow around a hemispherical body[13]. Consequently, we are confident about the reliability of such chemical model. Other chemical mechanisms like the one developed by Drummond[23] have been tested but results didn't match quantitatively experimental evidences.

Table 3.1 Chemical reaction mechanism used for the simulation. Units are in seconds, moles, cubic centimeters, calories, and Kelvins. The reaction constant of the kth chemical reaction is given in the form $k_k = A_k \cdot T^n \cdot e^{-E_k/RT}$. Third body efficiencies are equal to unity except:
Reaction (6): $H_2O/6.0$ Reaction (9): $H_2O/16.0$; $H_2/2.0$
Reaction (7): $H_2O/6.0$; $H_2/2.0$ Reaction (19): $H_2O/15.0$
Reaction (8): $H_2O/5.0$

No	Reaction	A	E	n
(1)	$H_2 + O_2 \rightleftharpoons HO_2 + H$	1.0×10^{14}	56,000	0
(2)	$H + O_2 \rightleftharpoons OH + O$	2.6×10^{14}	16,800	0
(3)	$O + H_2 \rightleftharpoons OH + H$	1.8×10^{10}	8,900	1
(4)	$OH + H_2 \rightleftharpoons H + H_2O$	2.2×10^{13}	5,150	0
(5)	$OH + OH \rightleftharpoons O + H_2O$	6.3×10^{12}	1,090	0
(6)	$H + OH \overset{M}{\rightleftharpoons} H_2O$	2.2×10^{22}	0	-2
(7)	$H + H \overset{M}{\rightleftharpoons} H_2$	6.4×10^{17}	0	-1
(8)	$H + O \overset{M}{\rightleftharpoons} OH$	6.0×10^{16}	0	-0.6
(9)	$H + O_2 \overset{M}{\rightleftharpoons} HO_2$	2.1×10^{15}	-1,000	0
(10)	$O + O \overset{M}{\rightleftharpoons} O_2$	6.0×10^{13}	-1,800	0
(11)	$HO_2 + H \rightleftharpoons OH + OH$	1.4×10^{14}	1,080	0
(12)	$HO_2 + H \rightleftharpoons H_2O + O$	1.0×10^{13}	1,080	0
(13)	$HO_2 + O \rightleftharpoons O_2 + OH$	1.5×10^{13}	950	0
(14)	$HO_2 + OH \rightleftharpoons H_2O + O_2$	8.0×10^{12}	0	0
(15)	$HO_2 + HO_2 \rightleftharpoons H_2O_2 + O_2$	2.0×10^{12}	0	0
(16)	$H + H_2O_2 \rightleftharpoons H_2 + HO_2$	1.4×10^{12}	3,600	0
(17)	$O + H_2O_2 \rightleftharpoons OH + HO_2$	1.4×10^{13}	6,400	0
(18)	$OH + H_2O_2 \rightleftharpoons H_2O + HO_2$	6.1×10^{12}	1,430	0
(19)	$H_2O_2 \overset{M}{\rightleftharpoons} OH + OH$	1.2×10^{17}	45,500	0

4. Numerical Analysis

The explicit finite difference of the Euler conservation equations (1) can be written as follows:

$$\begin{aligned}\Delta\hat{Q}_{i,j} = \hat{Q}^{n+1}_{i,j} - \hat{Q}^{n}_{i,j} = &-\frac{\Delta t}{\Delta\xi}\cdot(\tilde{E}^{n}_{i+1/2,j} - \tilde{E}^{n}_{i-1/2,j}) \\ &-\frac{\Delta t}{\Delta\eta}\cdot(\tilde{F}^{n}_{i,j+1/2} - \tilde{F}^{n}_{i,j-1/2}) \\ &-\Delta t\cdot\hat{H}^{n}_{i,j} + \textit{chemical source term}\end{aligned} \tag{7}$$

In recent years, many second-order-accurate shock-capturing finite-difference schemes for numerical computation of Eq. (7) have been developed. In the present work, we do not extensively discuss these schemes. One can readily refer to the literature to have a comprehensive view and discussion about existing high-performing numerical schemes[24,25]. In the coming section, we will briefly present the numerical method used in the code. Interested readers should refer to specific mentioned papers for detailed information.

We use a *second-order in time* and *second-order in space* accurate *explicit upwind TVD* algorithm using *flux extrapolation* and *Roe's averaging* method for the integration of the conservation equations of the fluiddyamics. The stiff chemical source terms are implicitely integrated using a splitting technique. This numerical coupling between the fluid–dynamic and chemical equations removes the need of subcycling the chemistry and is referred as a *semi-implicit* method. Sections 4.1 and 4.2 discuss the explicit numerical schemes used to solve Eq. (7) in absence of chemical source term. The implicit integration of the chemistry is discussed in section 4.3.

4.1. Upwind TVD Scheme

Generally speaking, most of the recently developed schemes are *Total Variation Diminishing* (TVD) for nonlinear, scalar, hyperbolic conservation laws. Defining the *total variation in x*, TV, of a discrete solution to a scalar conservation law by

$$TV(q) = \sum_i |\, q_{i+1} - q_i \,|\,,$$

a numerical scheme is said to be *total variation diminishing* if

$$TV(q^{n+1}) \le TV(q^n)\,.$$

TVD schemes are *monotonicity preserving* (no new local extrema in x can be created and the value of a local minimum is non-decreasing, the value of a local maximum is non-increasing). In other words, monotone profiles are preserved during the time evolution of the discrete solutions and no over- and undershoots will be created. This is a major requirement for simulation of supersonic flowfields with shock structures. Many explicit second-order accurate in time TVD schemes have been developed. The earliest is the Flux Corrected Transport (FCT) algorithm developed by Boris and Book[26] and generalized by Salezak[27]. It was the first time that the concept of non-linear flux limiting was introduced. The idea is to add to a first-order monotone solution a limited amount of the difference between second-order and first-order

fluxes. This difference is called *antidiffusive* flux and corrects the excessive dissipation of first-order schemes without creating unwanted oscillations, typical of the second-order schemes.

This approach has been adapted to new generations of second-order explicit TVD schemes, either through variable extrapolation or through flux extrapolation. As can be seen from Eq. (7), fluxes have to be calculated at the cell boundaries. One approach is to calculate state *variables* at the interfaces by *extrapolation* between the neighbouring cell averages. This variable extrapolation method is often referred in the literature as the MUSCL approach, standing for Monotone Upstream-centered Schemes for Conservation Laws after Van Leer[28]. The fluxes at the cell boundaries are then calculated from these values. The alternative option is to calculate the *fluxes* at the interfaces directly by *extrapolation* from their values in the cells. Since the fluxes are non-linear functions of the basic variables, flux extrapolation is not identical to variable extrapolation. We use flux extrapolation method, i.e. a non-MUSCL technique, and fluxes in Eq. (7) are thus directly calculated from their values in the cells by

$$\tilde{E}^n_{i+1/2,j} = \frac{1}{2} \cdot (\hat{E}^n_{i,j} + \hat{E}^n_{i+1,j} + R_{i+1/2} \cdot \Phi_{i+1/2}) \,. \tag{8}$$

Similar equations can be written for terms $\tilde{E}^n_{i-1/2,j}$, $\tilde{F}^n_{i+1/2,j}$, and $\tilde{F}^n_{i-1/2,j}$ of Eq. (7). The non-linear corrections $R_{i+1/2} \cdot \Phi_{i+1/2}$ are numerical dissipation terms. *Upwind TVD* scheme refers to the algorithm whereby the numerical dissipation term is upwind-weighted as opposed to the symmetric TVD scheme whereby the numerical dissipation term is centered. We use upwind TVD scheme. Note that this notion of upwind and symmetric TVD with limiting functions for the calculation of the dissipative terms no longer has its traditional upwinding meaning.

4.2. *Dissipation Term*

Computation of the correction terms $R_{i+1/2} \cdot \Phi_{i+1/2}$ is described in detail by Yee[29]. $R_{i+1/2}$ is the matrix whose columns are eigenvectors of $\partial\hat{E}/\partial\hat{Q}$ (ξ-direction) or $\partial\hat{F}/\partial\hat{Q}$ (η-direction) where $\hat{Q}$ is evaluated at the interfaces $i + 1/2$. The elements of matrix $\Phi_{i+1/2}$ denoted by $\phi^l_{i+1/2}$ ($l = 1, 4 + N$) are

$$\phi^l_{i+1/2} = \frac{1}{2} \left[\Psi(a^l_{i+1/2}) - \frac{\Delta t}{\Delta \xi} a^{l\ 2}_{i+1/2}\right] \cdot [g^l_{i+1} + g^j_i] - \Psi(a^l_{i+1/2} + \gamma^l_{i+1/2}) \cdot \alpha^l_{i+1/2}$$

with

$$g^l_i = S \cdot \max[0, \min(|\alpha^l_{i+1/2}|, S \cdot \alpha^l_{i-1/2})]$$

$$S = \operatorname{sign}(\alpha^l_{i+1/2})$$

$$\gamma^l_{i+1/2} = \frac{1}{2} \left[\Psi(a^l_{i+1/2}) - \frac{\Delta t}{\Delta \xi} a^{l\ 2}_{i+1/2}\right] \cdot \begin{cases} (g^l_{i+1} - g^l_i)/\alpha^l_{i+1/2} & \alpha^l_{i+1/2} \neq 0 \\ 0 & \alpha^l_{i+1/2} = 0 \end{cases}$$

$$\Psi(z) = \begin{cases} |z| & |z| \geq \epsilon \\ (z^2 + \epsilon^2)/2\epsilon & |z| < \epsilon \end{cases}$$

where ϵ is a small positive number[30].
$a^l_{i+1/2}$ are the eigenvalues of $\partial\hat{E}/\partial\hat{Q}$ (in the ξ direction). $\alpha^l_{i+1/2}$ is the difference of the conserved variables and are elements of (in ξ-direction):

$$\hat{\alpha}_{i+1/2} = R^{-1}_{i+1/2}(\hat{Q}_{i+1,j} - \hat{Q}_{i,j})$$

The *limiter* function g^l_i is used to prevent generation of numerical oscillations, under- and overshoots. Various type of limiters has been tested; for the property and usage of these functions the reader can refer to Yee's work[29].
In such a formulation, the scheme is second-order in time and space.
As mentioned above, the calculation of $R_{i+1/2}$ and $\Phi_{i+1/2}$ requires knowledge of the state quantities at the interfaces. A simple interpolating way which immediately comes to mind is the arithmetic average

$$\hat{Q}_{i+1/2,j} = \frac{1}{2}(\hat{Q}_{i,j} + \hat{Q}_{i+1,j}) .$$

But this does not take into account the non-linear features of the flowfield. To approximate the Riemann solver at the interfaces, Roe suggested to use state variables replaced by their average weighted by the square root of the densities $\Lambda_{i+1/2} = \sqrt{\rho_{i+1}/\rho_i}$ [25,29,31]:

$$\begin{aligned} \rho_{i+1/2} &= \sqrt{\rho_{i+1}\rho_i} = \Lambda_{i+1/2}\rho_i \\ u_{i+1/2} &= \frac{(u\sqrt{\rho})_{i+1} + (u\sqrt{\rho})_i}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} = \frac{\Lambda_{i+1/2}u_{i+1} + u_i}{\Lambda_{i+1/2} + 1} \\ &\cdots \end{aligned} \tag{9}$$

Roe has shown that this averaging method has the computational advantage of perfectly resolving stationary discontinuities but is only applicable to a perfect gas[31].

4.3. *Semi Implicit Integration*

Semi-implicit TVD-type scheme for the transformed system of Eq. (1), including the stiff chemical source terms, can be written as[9,32]

$$\begin{aligned} D^n_{i,j}\Delta\hat{Q}_{i,j} = &-\frac{\Delta t}{\Delta\xi}\cdot(\tilde{E}^n_{i+1/2,j} - \tilde{E}^n_{i-1/2,j}) - \frac{\Delta t}{\Delta\eta}\cdot(\tilde{F}^n_{i,j+1/2} - \tilde{F}^n_{i,j-1/2}) \\ &- \Delta t\cdot\hat{H}^n_{i,j} + \Delta t\cdot\hat{S}^n_{i,j} \end{aligned} \tag{10}$$

$$D^n_{i,j} = I - \theta\cdot\Delta t\cdot\frac{\partial\hat{S}}{\partial\hat{Q}} \tag{11}$$

Matrix D is called the time-rescaling matrix. The parameter θ is in the range $0 \leq \theta \leq 1$; for any $\theta \neq 0$, the source terms are treated implicitly. We use $\theta = 1/2$. The Jacobian of the source terms, $\partial \hat{S}/\partial \hat{Q}$ is given by

$$\partial \hat{S}/\partial \hat{Q} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \dfrac{\partial \omega_1}{\partial \rho} & \dfrac{\partial \omega_1}{\partial \rho u} & \dfrac{\partial \omega_1}{\partial \rho v} & \dfrac{\partial \omega_1}{\partial E} & \dfrac{\partial \omega_1}{\partial \rho_1} & \cdots & \dfrac{\partial \omega_1}{\partial \rho_N} \\ \vdots & & & \vdots & \vdots & & \vdots \\ \dfrac{\partial \omega_N}{\partial \rho} & \dfrac{\partial \omega_N}{\partial \rho u} & \dfrac{\partial \omega_N}{\partial \rho v} & \dfrac{\partial \omega_N}{\partial E} & \dfrac{\partial \omega_N}{\partial \rho_1} & \cdots & \dfrac{\partial \omega_N}{\partial \rho_N} \end{vmatrix} \tag{12}$$

where the elements of the matrix have to be calculated from Eq. (6) through

$$\frac{\partial \omega_i}{\partial \rho} = \frac{\partial \omega_i}{\partial T} \cdot \frac{\partial T}{\partial \rho}, \quad \cdots \quad, \quad \frac{\partial \omega_i}{\partial \rho_j} = \frac{\partial \omega_i}{\partial T} \cdot \frac{\partial T}{\partial \rho_j} \tag{13}$$

5. Code Structure and Guidelines

In the present section, the actual implementation of the algorithm is described. The computer codes use standard FORTRAN-77 language. The aim of this section is to clarify the overall code structure as well as the various input-parameters. A detailed description of each subroutine and variable is not covered although every entry point the user can modify is mentioned. This section has to be read in close connection with the computer code. The first item describes the nonreactive flow computation. In the second item, the implementation of the chemistry source-terms is explained. Details about the grid making are given in the last paragraph.

5.1. Nonreactive Code

The nonreactive code integrates Eq. (1) without any chemical source term. The gas is composed of oxygen O_2 and nitrogen N_2. The following initial flow conditions are freely choosen by the user:

1. Initial presure and temperature.
2. Inflow velocity and velocity of the body.
3. Ratio O_2/N_2.

The code is designed to converge to a steady flowfield, although initial conditions and boundary geometry may prevent from obtaining a quite stable flow field. It is the user's duty to analyse the convergence process. Output of nonreactive calculation can be readily and usefully be used as initial flow field for a reactive simulation.

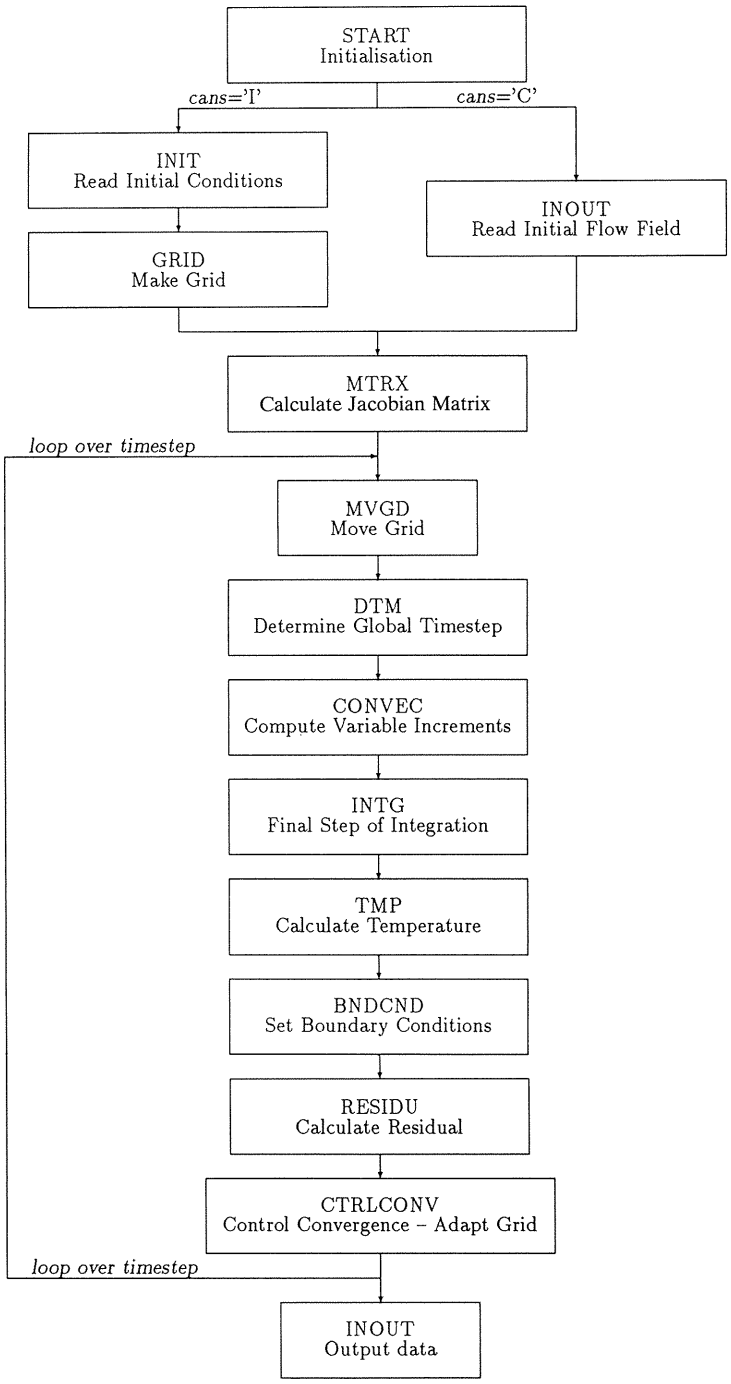


Fig. 5.1. Overall flowchart indicating the subroutines involved in the non reactive simulation. Subroutines are indicated in capital letters. Some subroutines mentioned can be skipped according to initial input.

5.1.1. *Overall Description*

The overall flow diagram is shown in Fig. 5.1. Routine START reads the problem to determine whether or not an initial flowfield has to be generated by calling subroutine INOUT. If the calculation starts with new initial conditions, INIT is called to read all initial physical and numerical parameters. The mesh system is generated by routine GRID.
Subroutine MTRX calculates the metrics for the general coordinate system $\xi - \eta$ (variables *dmtx(j,k,1 to 4)*) and the grid Jacobian J (variable *dmtx(j,k,5)*).
The integration process starts with MVGD which implements the body velocity. To be compatible with the program, the body velocity, *dub*, must be given in the x-direction.
Routine DTM calculates the timestep Δt according to the Courant condition

$$\Delta t = \frac{CFL}{max(U + V + a(\xi_x^2 + \xi_y^2)^{1/2} + a(\eta_x^2 + \eta_y^2)^{1/2})},$$

where *CFL* is the Courant number. A typical value of *CFL* is 0.5, although higher values could be used for nonreactive computation.

The actual integration of the conservation equations is carried out by subroutine CONVEC. This routine calculates the increment of all physical quantities over the entire timestep Δt. This is discussed in some details in 5.1.4. The implementation of the calculated increments occurs in INTG. The latter subroutine ensures also positivity for the species densities and normalizes the total density according to the sum of species densities.

Subroutine TMP calculates the temperature taking into account the temperature dependence of the heat capacity (see 5.1.5.).
The boundary conditions are set by calling BNDCND (see 5.1.6.).
Control of the convergence is based on the calculation of the residual (subroutine RESIDU) and on successive modifications of the accuracy parameters (subroutine CTRLCONV). This procedure is explained in detail in 5.1.7. Subroutine CTRLCONV controls also the adaptive grid procedure.
Subroutines MVGD to CTRLCONV represent the core of the loop over time. The calculation is over when convergence is achieved or when the maximum number of timesteps *lend* has been run, whatever comes first. Note that an internal clock prevents computation for running over CPU-time. The user can check and control the maximun allocated CPU-time in the main program.
The program ends with the printing and storage routine INOUT.

5.1.2. *Starting Conditions*

Initialisation of the computation starts by reading the *cans* variable. The user decides to carry out new calculation (*cans* = 'I') or to continue a calculation using an existing flowfield (*cans* = 'C'). Using the 'I'-variable will result in reading the initial conditions (calling INIT) and making the numerical grid (calling GRID). When the 'C'-variable is used, the program reads the geometry and existing flowfield from file *name1* (calling INOUT) and performs further calculation.

5.1.3. *Input Parameters*

Subroutine INIT provides the program with the required initial parameters. They are read in a separate file. Depending on the computer used, this input-file can be combined with the run-file of the main program. Compiling, running, and I/O procedures are computer-type dependent and are not discussed here. The actual inputs required from the users are:

cans: character*1; Starting conditions. Possible values are 'I' (start a new calculation) or 'C' (continue previous calculation, read data from *name1*)

bavt: character*3; Determine the type of average in the TVD-scheme. Possible values are 'ROE' (ROE-average) or 'LIN' (linear average).

baod: character*1; Order of the TVD-integration. Possible values are '1' (first-order) and '2' (second-order). Typical value is '1'. The program will automatically upgrade to '2' during convergence process.

btst: character*2; Determine the type of timestep. Possible values are 'GT' (global timestep) or 'LT' (local timestep).

lpln: character*1; Polynomial type for the thermodynamic data. Possible values are '1' or '2'.

ddlt: real; initial entropy factor. A typical value is 0.8. It will gradually decrease down to 0.15 during the convergence process.

dcfl: real; CFL number.

lend: integer; Last timestep.

linf: integer; Frequency, expressed in term of timestep, for printing info-messages.

lfgr: integer; Frequency, in term of timestep, for saving physical quantities.

baxs: character*1; Axisymmetric ('Y') or two dimensional ('N').

bgrd: character*1; Control the call to the adaptive grid algorithm. Possible values are 'N' (numerical grid will not be adapted) or 'Y' (perform adaptive grid according to the convergence process)

aercn: real; Convergence criterion. Calculation ends when residual < *aercn.*

dcnd: real; When the upper limit of the computational domain is an horizontal slab, *dcnd* expresses its boundary condition. Possible values are '1.0' (outflow) and '−1.0' (reflecting wall).

name1: character*18; Filename for saving basic quantities and parameters. This file is used to restart a calculation or to start a reactive flow calculation.

name2: character*18; Filename for saving grid coordinates.

name3: character*18; Filename for saving physical quantities. This file has to be used for post-processing.

name4: character*18; Filename for saving miscellaneous parameters. Currently save the residual at regular step-interval. The purpose of this file can be changed freely by the user.

anm(1): real; Number of mol O_2.

anm(2): real; Number of mol N_2.

jj: integer; Number of numerical cells parallel to the body surface.

kk: integer; Number of numerical cells normal to the body surface.

dui: real; Inflow velocity (m/s).

dub: real; Velocity of the solid body (m/s).

dprs: real; Initial pressure (atm).

dt: real; Initial temperature (K).

ards to *jj4:* Geometry parameters; See section Grid Geometry.

When option *continue* is used (*cans* = 'C'), the user keeps control on the above parameters up to *name4.* Parameters from *anm(1)* to *jj4* are then unalterable and are read from file *name1.*

An example of a typical input-file is given in Table 5.1. The first six lines are specific to the VP2600-computer of the Nagoya University and must be properly modified by the user when necessary.

Table 5.1. Sample of input-file.

```
//W49237A JOB ,CLASS=U,REGION=6144K
//EXEC FORTCLG,PARM.FORT='DOUBLE,S,VP,AE,SIZE(MAX)',
VP=ON, FREGION=40M
//FORT.SYSIN DD DSN=W49237A.LEFSOS.DIR(RAMNEW),DISP=SHR
//FORT.SYSINC DD DSN=W49237A.LEFSOS.DIR,DISP=SHR
//GO.SYSIN DD *
I,ROE,1,GT,2                                  (CANS : BAVT : BAOD : BTST : LPLN)
0.80,0.5,20000                                      (DDLT : DCFL : LEND)
0,0                                                      (LINF : LFGR)
Y,N                                                      (BAXS : BGRD)
-5.0,-1.0                                               (AERCN : DCND)
W49237A.LEFSOS.RES                                             (NAME1)
W49237A.LEFSOS.GRD                                             (NAME2)
W49237A.LEFSOS.FLW                                             (NAME3)
W49237A.LEFSOS.MIS                                             (NAME4)
1.0,3.76                                               (ANM(1): ANM(2))
85, 45                                                          (JJ : KK)
2400.,0.,0.100,298.                                (DUI : DUB : DPRS : DT)
0.0035,0.002,0.035,0.04,0.0012          (ARDS : ARDS2: AL1 :AL2 :AL3)
0.004,70.0,5.0                                  (AH2 : ALPHA: ALPHA2)
15,30,10,30                                      (JJ1 : JJ2 : JJ3 : JJ4)
//
```

5.1.4. *Integration of Euler Equations*

Eq. (7) is implemented in subroutines CONVEC and INTG. Flowchart of routine CONVEC is given in Fig. 5.2. The routine is run twice for each direction ξ and η successively. First subroutine FLUX is called to calculate the fluxes

$$\frac{1}{2} \cdot (\hat{E}^n_{i,j} + \hat{E}^n_{i+1,j})\ (\xi - \textit{direction}) \quad \text{and}$$

$$\frac{1}{2} \cdot (\hat{F}^n_{i,j} + \hat{F}^n_{i,j+1})\ (\eta - \textit{direction})$$

in the general coordinate system $\xi - \eta$ (see Eq. (8)).
The state variables needed for the computation of the dissipative terms ($R_{i+1/2,j} \cdot \Phi_{i+1/2j}$ in Eq. (8)) are calculated according to the selected averaging method: Roe's average by calling routine RAVG or linear method by calling LAVG. These average values are used in routine NMUSC which calculates the dissipative terms and the corrected fluxes at the interfaces, i.e. $\tilde{E}^n_{i+1/2,j}$, $\tilde{E}^n_{i-1/2,j}$, $\tilde{F}^n_{i+1/2,j}$, and $\tilde{F}^n_{i-1/2,j}$, (Eq. (8)). A temporary increment $\Delta\hat{Q}_{i,j}$ (variable *drhs(i,j,k)* in the code) is calculated. When both integrations in ξ and η direction are carried out, the axisymmetric term $\hat{H}^n_{i,j}$ (Eq. (7)) is calculated and added to the pending increment $\Delta\hat{Q}_{i,j}$.
The finite difference integration

$$\hat{Q}^{n+1}_{i,j} = \hat{Q}^n_{i,j} + \Delta\hat{Q}_{i,j}$$

is finally performed in subroutine INTG. Moreover, INTG makes sure that species densities remain positive.

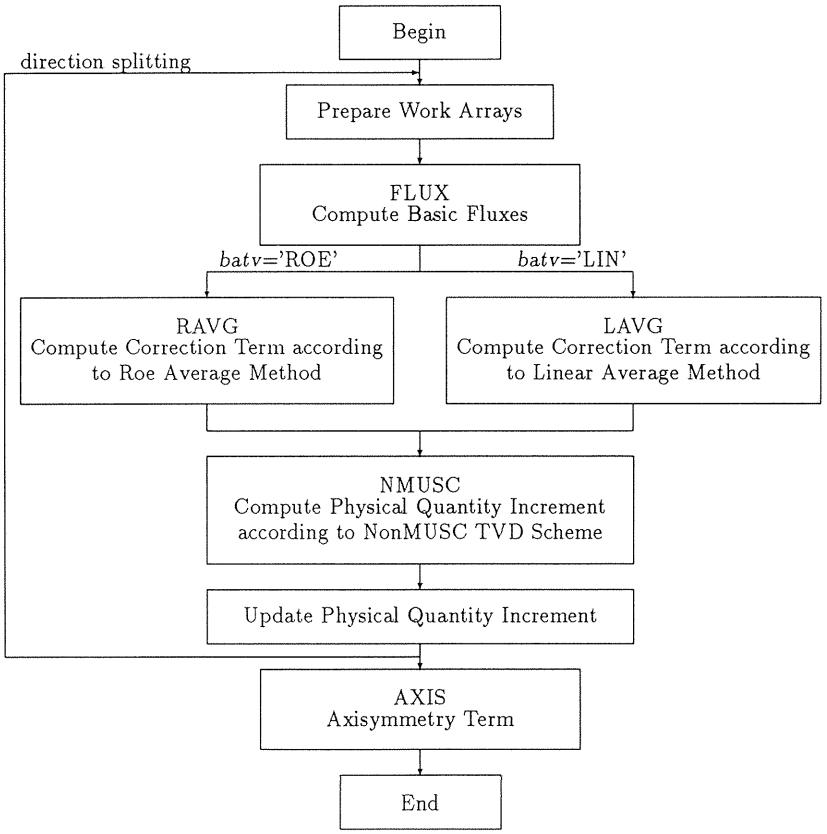


Fig. 5.2. Flowchart of subroutine CONVEC. The subroutine calculates the change in physical quantities over the entire timestep.

5.1.5. *Calculation of Temperature*

The temperature is calculated in subroutine TMP. Using polynomial form for the thermodynamic data and Newton-Raphson method for numerical analysis, the temperature is computed from the knowledge of the internal energy. No values less than the initial temperature are allowed to eliminate numerical undershots. The thermodynamic data provided by the BLOCKDATA are two sets of 7 coefficients for temperatures ranging from 300 K to 1000 K and from 1000 K to 5000 K, respectively. They give the enthalpy and entropy for each species under the following form:

$$\frac{h^o}{RT} = a_1 + a_2\frac{T}{2} + a_3\frac{T^2}{3} + a_4\frac{T^3}{4} + a_5\frac{T^4}{5} + \frac{a_6}{T}$$

$$\frac{s^o}{R} = a_1 lnT + a_2 T + a_3\frac{T^2}{2} + a_4\frac{T^3}{3} + a_5\frac{T^4}{4} + a_7$$

h^o and s^o refer to enthalpy and entropy at a reference state of one atmosphere. Coefficients a_i are read from blockdata IDATA.

5.1.6. *Boundary Conditions*

The overall boundary conditions are shown in Fig. 2.1. The right boundary (k=*kk*) imposes the permanent initial inflow condition and the left boundary (j=*jj*) represents outflow condition. When the upper domain is limited by an horizontal slab, the user can select either outflow or reflecting wall condition by equating *dcnd* input-parameter to '1' or '-1', respectively. When the geometry is such that there is no horizontal upper boundary, parameter *dcnd* is meaningless. Outflow condition is characterized by equal physical quantities through one guard cell. Reflecting wall condition is characterized by first order extrapolation, free slip condition. On a similar way, the symmetry is calculated by a 1st-order extrapolation, i.e. symmetry is ensured using one numerical guard-cell.
The body surface is characterized by a slip and adiabatic condition as shown in Fig. 5.3.

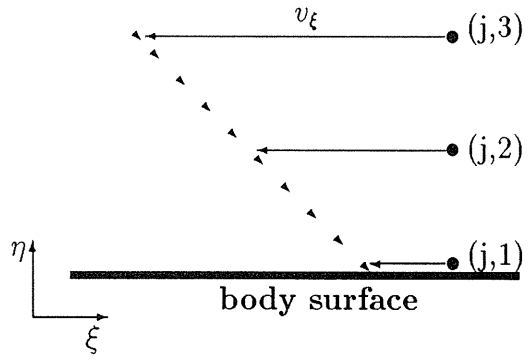


Fig. 5.3. Slip and adiabatic boundary condition on the body surface.

5.1.7. *Convergence*

Control over the convergence process starts with subroutine RESIDU which calculates the residual *derr*. The residual is theoretically the average absolute change in total density between two successive timesteps. As a matter of fact, 'residual' (variable *derr*) will refer from now on as the logarithm of the above defined residual. Actually, routine CTRLCONV keeps the overall convergence process under control. As the residual becomes small, i.e. as convergence approaches, CTRLCONV tightens the convergence criterion and the accuracy parameter, as shown in the Flowchart 5.4. When the residual becomes smaller than −3.5, the entropy factor *ddlt* is reduced in two steps to its lowest value of 0.15. In the meantime, second order TVD-scheme is automatically set (*boad* = '2'). Subroutine ADPGRD is one more time

called when residual reaches −4.5 and from then, calculation continues up to the final convergence condition *residual* < *aercn.*

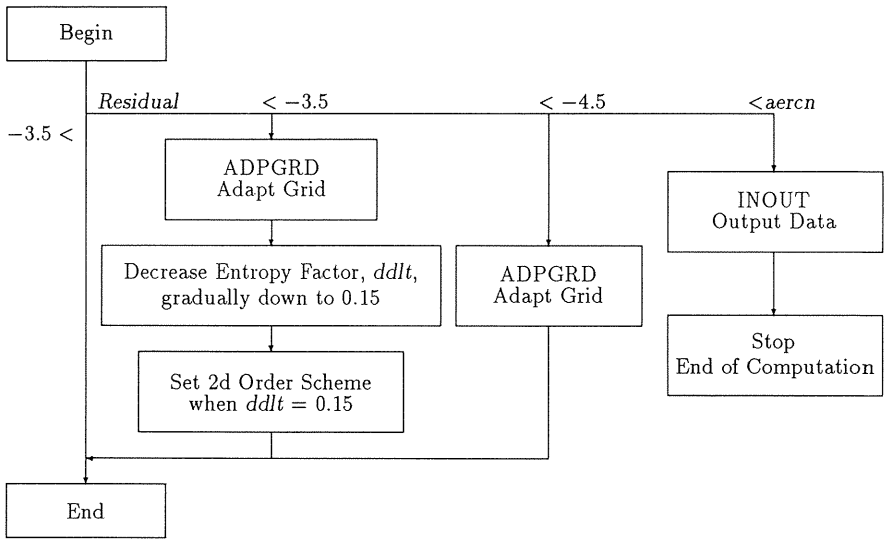


Fig. 5.4. Flowchart of subroutine CTRLCONV. The value of *aercn* is given by the initial input. A typical value ensuring correct convergence is −5.

5.1.8. Adaptive Grid

The routine ADPGRD is called according to the convergence process as described in the above item and only if the call to the routine is on (*bgrd* = 'Y'). The aim of this subroutine is to focus the middle of the grid into the leading shock and to redistribute the grid points on both sides of it. One third of the new grid points are redistributed in the inflow domain when the remaining two thirds of the points are relocated behind the bow shock where fine grid-resolution is required. The user can change this distribution by modifying the argument *iad* when calling ADPGRD. Because the latter parameter is not an input-variable, the change has to be carried out in the subroutine CTRLCONV. Flow chart 5.5. gives the main parts of routine ADPGRD.

The routine starts by defining the shock-line in order to settle the grid on either side of it. When a geometry including an horizontal upper slab with reflecting boundary condition is used, the routine can not scan properly the numerical domain and can not define a single shock-line. Therefore, when such a geometry is used, routine ADPGRD should not be called.

When the new grid system is determined, values for the physical quantities are calculated by linear interpolation. Call to MTRX takes care of the coordinate transformation.

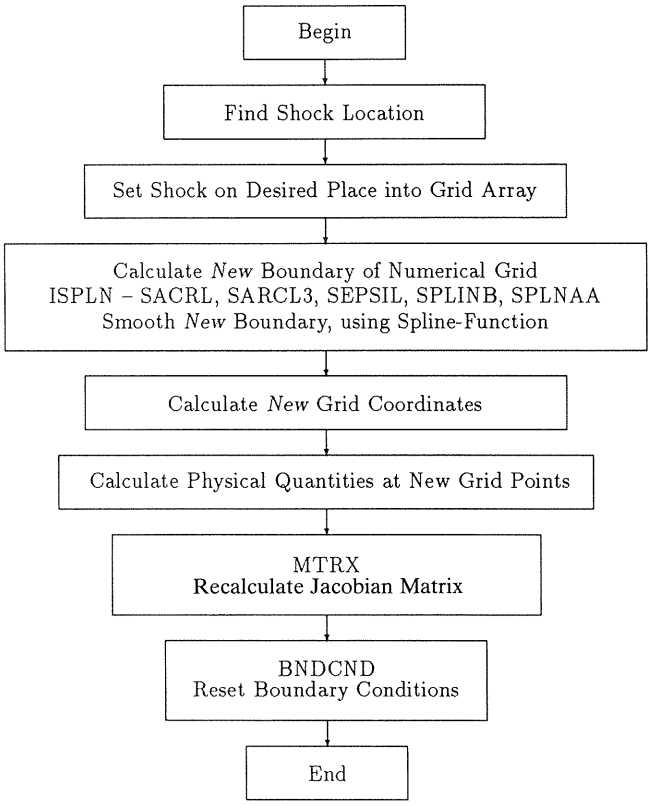


Fig. 5.5. Flowchart of subroutine ADPGRD. The subroutine starts with the *old* values for grid coordinates and physical quantities and brings out *new* values.

5.1.9. Output Data

Output is twofold. First, basic physical quantities (density, momentums, energy, species densities) and basic numerical and physical parameters (grid coordinates, timestep, initial conditions) are saved in the file *name1*. This file allows the user to continue the calculation using the current flowfield as starting condition. Second, various data are saved for post-processing. Grid coordinates are saved in file *name2*. Total density, pressure, temperature, velocities along x- and y-axis, and Mach number are saved in file *name3*. The file *name4* is open early in the program and closed at the end of the calculation and is designed for the user to save whatever variables he wants. Changes in output can be carried out in subroutine INOUT.

5.1.10. Error Messages

Beside standard error messages provided during compilation or execution of Fortran programs (about these messages, see reference Fortran manual), specific messages and advisory warnings are provided. They are:

ERROR No 1: The number of cells of all sub-parts of the grid does not match the total number of cells ($\Sigma jj_i \neq jj$).

ERROR No 2, #n: There are lines in the grid where no shock can be detected (η-direction). #n refers to the line involved. Occur in ADPGRD-subroutine when searching for the location of the shock. Note that the shock position is given by the grid cell where the total density becomes greater than twice the upstream total density.

ERROR No 3, #n: The leading shock has reached the boundary at line #n. Occur in ADPGRD-subroutine when searching for the location of the shock.

ERROR No 4: Invalid input parameters.

ERROR No 5: Stand-off distance too large. The stand-off distance is larger than the input parameter *al3*.

ERROR No 6: Timestep too small.

ERROR No 7: Out of grid range. Maximum ranges can be checked and modified in the COMMON–file.

5.2. Reactive Code

The reactive code XXX2 is build using same general frame as for the nonreactive code XXX1. In a first section, we describe the overall structure of Code XXX2 emphasizing the differences between both codes. Next, initial and input parameters are discussed. The user can readily refer to section 5.1.4 and flowchart 5.2 for a description of the Euler-equations integration. Calculation of the temperature and implementation of the boundary condition are also quite similar for both codes and are not discussed further. No adaptive gridding is performed in the reactive code; therefore, the convergence follow-up, carried out in subroutine RESIDU, only calculates the residual and checks whether or not it becomes less than the *aercn*–variable.

5.2.1. Overall Description

The general chart describing the structure of the reactive code is shown in Fig. 5.6. Subroutine START is discussed in more details below. The Euler conservation equations are solved separately from the chemical source terms in subroutine CONVEC. This routine is similar to the CONVEC routine in the nonreactive code. It computes the increments over the entire timestep of all basic physical quantities: total density, momenta, total energy, and species densities. The chemical reaction rates are integrated in subroutine REACT. This routine can be skipped by equating parameter *bcrt* to 'N'. In that case, the computation is a simulation of frozen flow containing 9 nonreacting species. Section 5.2.4 deals with detailed discussion of subroutine REACT. Normalizing total density is carried out in routine INTG after performing the final finite difference integration. Note that the grid remains unchanged during all the computation. No adaptive grid algorithm is provided. It is assumed that the ideal grid configuration has been achieved during the nonreactive computation.

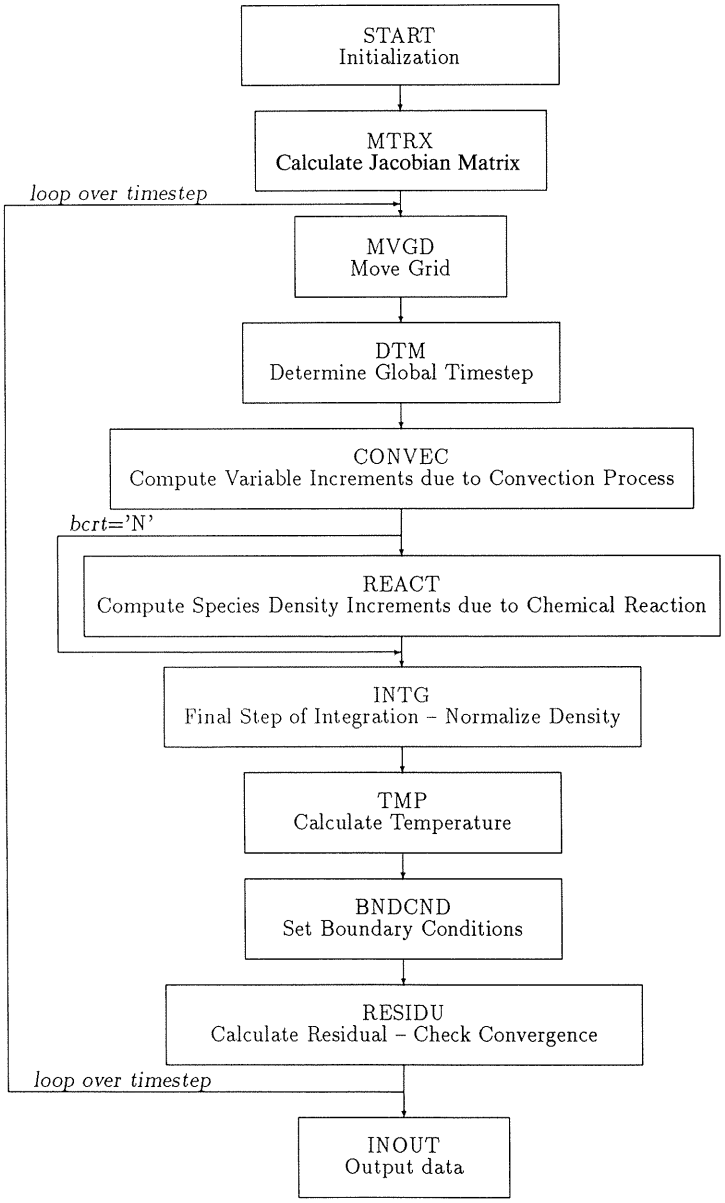


Fig. 5.6. Overall flowchart of the reactive code XXX2.

5.2.2. *Starting Procedure*

Ways to start the computation are twofold: one can (1) start a new reactive computation using the steady flowfield of a non reactive simulation or (2) continue a reactive computation. Description of the START subroutine is shown in Fig. 5.7.

When using the output of a 2-species nonreactive calculation (performed with Code XXX1), the program starts by reading general parameters, grid parameters (from file *name4*) and two-species flowfield data (from file *name4*). Subroutine SETRCT scans the entire numerical domain to locate the front shock and initializes the new species densities in front of it. The total energy is modified according to the internal energy content of the new chemical composition. If, during this scanning process, no shock can be detected along one of the η-lines, an error message is printed and the computation fails.
When proceeding with a reactive computation, flowfield data are read from *name4* through subroutine INOUT. The user has the opportunity to modify the freestream velocity by equating *cvl*-parameter to 'Y'. In that case, a new incoming velocity is implemented in front of the leading shock and momentum and total density of energy are changed accordingly. Morever, this option enables the user to modify the incoming chemical composition. When one wants to change slightly the upstream conditions, it is particularly computer-time saving to use this option instead of computing a new cycle of nonreactive—reactive simulations.

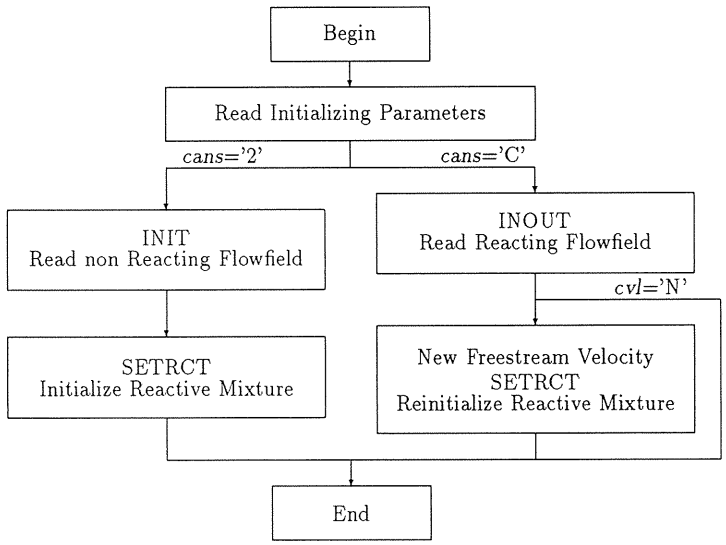


Fig. 5.7. Flow chart of subroutine START. Parameter *cans* determines the type of initial flowfield used in the calculation.

5.2.3. Input Parameters

A set of input parameters are required from the user whatever the starting option *cans* is. They are

cans: character*1; Starting conditions. Possible values are '2' (start a new calculation using datafile from a nonreactive calculation performed with two species mixture, read data from *name4*) or 'C' (continue previous reactive computation, read data from *name4*)

batv: character*3; Determine the type of average in the TVD-scheme. Possible values are 'ROE' (ROE-average) or 'LIN' (linear average).

baod: character*1; Order of the TVD-integration. Possible values are '1' (first-order) and '2' (second-order). Typical value is '2'.

btst: character*2; Determine the type of timestep. Possible values are 'GT' (global timestep) or 'LT' (local timestep).

lpln: character*1; Polynomial type for the thermodynamic data. Possible values are '1' or '2'.

ddlt: real; initial entropy factor. A typical value is 0.15 to match the lowest value reached in the nonreactive computation.

dcfl: real; Courant number.

lend: integer; Last timestep.

inf: integer; Frequency, expressed in terms of timestep, for printing info-messages.

lfgr: integer; Frequency, in terms of timestep, for saving physical quantities.

baxs: character*1; Axisymmetric ('Y') or two dimensional ('N').

bcrt: character*1; Turn on the chemical reaction. Possible values are 'N' (the computation will not allow the chemical reaction to develop) or 'Y' (integrate the chemical rates)

cvl: character*1; Change freestream velocity ('Y') or not ('N'). When *cvl* = 'N', *wdui* is meaningless.

aercn: real; Convergence criterion. Calculation ends when residual < *aercn.*

dcnd: real; When the upper limit of the computational domain is an horizontal slab, *dcnd* expresses its boundary condition. Possible values are '1.0' (outflow) and '−1.0' (reflecting wall).

wdui: real; New freestream velocity (m/s).

name1: character*18; File for saving basic quantities and parameters. This file is used when the user wants to temporarely stop a calculation and restart it later on.

name2: character*18; File for saving grid and flow variables.

name3: character*18; File for saving miscellaneous physical and numerical quantities. Currently, the residual is saved at regular step-interval. The purpose of this file can be changed freely by the user.

name4: character*18; Input file containing the initial flowfield. It can be a reactive flowfield (*cans* must then be equal to 'C' and the datafile is called by subroutine INOUT) or a nonreactive flow (*cans* must then be equal to '2' and the datafile is called by subroutine INIT).

anm(1 to 9): real; Number of mol of H_2, O_2, *H*, *O*, *OH*, H_2O, HO_2, H_2O_2 and N_2, respectively.

When initializing the flowfield with steady nonreactive data (*cans* = '2'), it is recommended but not mandatory that input parameters like average type (*bavt*), order of integration (*baod*), polynomial type (*lpln*), and entropy factor (*ddlt*) match their values used during the nonreactive calculation. Similarly, for consistency reason, geometry parameter *baxs* and boundry parameter *dcnd* should remain unchanged during cyclical calculations. A typical value for the Courant number is 0.5, like for the nonreactive case. Nevertheless, when very fast energy release occurs, values around 0.3 can be required to perform trouble-free and correct integration. When proceeding with a reactive calculation (*cans* = 'C'), variables *anm*(1 to 9) are not read unless *cvl*-parameter is set to 'Y'. Note that *bcrt*-variable is a convenient parameter to turn off the chemical reaction. It enables the user to perform computations using correct thermochemistry for multi-species non-reacting flow (mixtures containing any ratio of hydrogen, oxygen, nitrogen, steam).

A typical input-file is given in Table 5.2. Command statements specific to the Fujitsu VP2600-computer are not shown.

Table 5.2. Sample of input-file for reactive flow computation.

//	
C,ROE,2,GT,2	(CANS : BAVT : BAOD : BTST : LPLN)
0.15,0.5,20000	(DDLT : DCFL : LEND)
0,0	(LINF : LFGR)
Y,Y,N	(BAXS : BCRT : CVL)
-5.0,-1.0,2300.	(AERCN : DCND : WDUI)
W49237A.LEFSOD.RES	(NAME1)
W49237A.LEFSOD.FIG	(NAME2)
W49237A.LEFSOD.MIS	(NAME3)
W49237A.LEFSOS.RES	(NAME4)
2., 1., 0., 0., 0., 0., 0., 0., 7.	(ANM(1 to 9))
//	

5.2.4. *Integration of Chemical Rates*

The integration of the chemical rates over the entire numerical domain is performed by routine REACT, called from the main program. Actual integration in individual numerical cells is carried out by subroutine CHEM. The flowchart of the latter is shown in Fig. 5.8. The subroutine calculates successively (1) the chemical reaction rates according to the mechanism described in Table 3.1, (2) the reverse reaction rates based on thermodynamic equilibrium condition, (3) species production rates (Eq. (6)), and a temporary increment $\Delta \hat{Q}_{i,j}$ (right-hand side of Eq. (10)). The routine calculates next some of the derivatives necessary to compute elements of the chemical source Jacobian (Eq. (12)), namely

$$\frac{dk_{k,for}}{dT} \quad \text{and} \quad \frac{dk_{k,back}}{dT},$$

Subroutine CHEMSJ is called for computing the elements of Jacobian (12). CHEMSJ computes all elements of Eq. (12): variables *daa*(*j, i, l = 1 to 4*) account for the derivative of ω_i with respect to the four global conserved quantities (ρ, ρu, ρv, and E) and variables *dbb*(*j, i, l = 1 to N*) for the derivative of ω_i with respect to the species densities ρ_l:

$$daa(j, N, 4) = \begin{vmatrix} \frac{\partial \omega_1}{\partial \rho} & \frac{\partial \omega_1}{\partial \rho u} & \frac{\partial \omega_1}{\partial \rho v} & \frac{\partial \omega_1}{\partial E} \\ \vdots & \cdots & \cdots & \vdots \\ \frac{\partial \omega_N}{\partial \rho} & \frac{\partial \omega_N}{\partial \rho u} & \frac{\partial \omega_N}{\partial \rho v} & \frac{\partial \omega_N}{\partial E} \end{vmatrix}$$

$$dbb(j, N, N) = \begin{vmatrix} \frac{\partial \omega_1}{\partial \rho_1} & \cdots & \frac{\partial \omega_1}{\partial \rho_N} \\ \vdots & \cdots & \vdots \\ \frac{\partial \omega_N}{\partial \rho_1} & \cdots & \frac{\partial \omega_N}{\partial \rho_N} \end{vmatrix}.$$

Index j stands for grid index. The Jacobian source term (Eq. (12)) is thus computed and stored under the following form:

$$\partial \hat{S}/\partial \hat{Q} = \begin{vmatrix} 0(j, 4, 4) & 0(j, 4, N) \\ daa(j, N, 4) & dbb(j, N, N) \end{vmatrix} \tag{14}$$

Routine REVSJ reverses matrix *dbb* and routine PIMP completes the inversion of Jacobian (14) by inverting elements of *daa*. Final update of the increment $\Delta \hat{Q}_{i,j}$ is achieved in PIMP according to the time-rescaling matrix Eq. (11) and the semi-implicit finite difference Eq. (10).

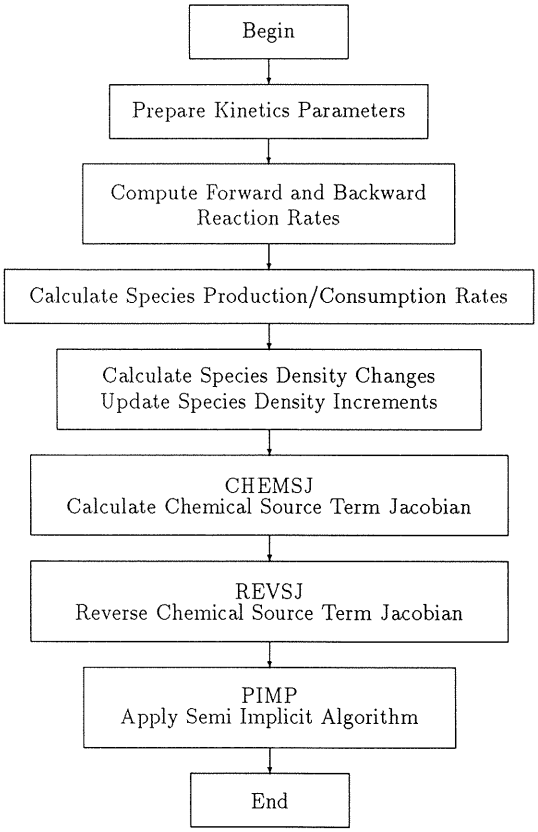


Fig. 5.8. Flowchart of subroutine CHEM. The subroutine calculates the change in species density over the entire timestep due to the chemical reaction. Subroutine CHEM is called by sub REACT in main program.

5.2.5. *Output Data*

Output data are
(1) Unformatted basic physical quantities which can be used to proceed with the calculation later on. These data are saved in file *name1*.
(2) Unformatted variables for graphics post-processing and data analysis. Currently grid coordinates, pressure, temperature, rate of H_2O production, and Mach number are saved in file *name2*. The user can modify this sequence freely in subroutine INOUT.
(3) File *name3* is at user's disposal for any variables. In the present code, the residual is recorded at regular timestep interval.

5.2.6. *Error Messages*

Similarly to Code XXX1, error messages are provided to the user. Possible messages are described below and always results in the computation to be stopped.

ERROR No 2, #n: No shock can be detected at grid line #n. Occur in subroutine SETRCT when looking for the location of the shock.
ERROR No 3: The leading shock has reached the upper right corner of the grid (*jj*-1, *kk*-1). Occur in subroutine TMP when scanning the temperature field.
ERROR No 5: Stand-off distance becomes too large. It means that the stand-off distance is larger than the input parameter *al3* or that the combustion has reached the lower right corner of the numerical domain (1, *kk*-1).
ERROR No 6: Timestep too small.

5.3. *Grid Geometry*

5.3.1. *Geometry Description*

The general geometry for the numerical grid is the overall design of a RAMAC device (Fig. 5.9). The configuration consists of a conical nose smoothly connected to the main conoid body of the projectile. By varying the parameters of each section, different configurations can be designed. For instance, when *jj3* and *jj4* are equated with 0.0, a conical blunt body is generated (Fig. 5.10).

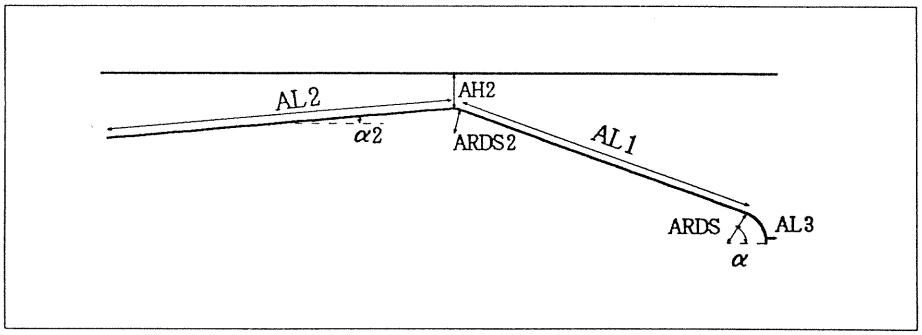


Fig. 5.9. Input parameters required to describe the general grid geometry.

5.3.2. *Input Parameters*

The input-parameters for describing the grid geometry go from *ards* to *jj4* as shown in Table 5.1. The definition of each parameter can be found in Fig. 5.9. All geometry parameters are *real.* Numbers of grid points jj_i are *integer.* The user must make sure that $\Sigma jj_i = jj$. An example of gridding for a conical body without afterbody is shown in Fig. 5.10. The user has to specify the following parameters:

ards: radius of the cone tip (m).
ards2: radius of the connection portion (m).
al1: length of the conical nose (m).
al2: length of the conoid aft (m).
al3: distance in front of the tip (m).
ah2: distance between the largest diameter of the body and the horizontal upper slab (m).
alpha: half tip angle (deg.).
alpha2: deflection angle of the main body (deg.)
jj1: number of numerical grid points along the blunt tip.
jj2: number of numerical grid points along the cone surface.
jj3: number of numerical grid points in the connection portion.
jj4: number of numerical grid points along the conoid surface.

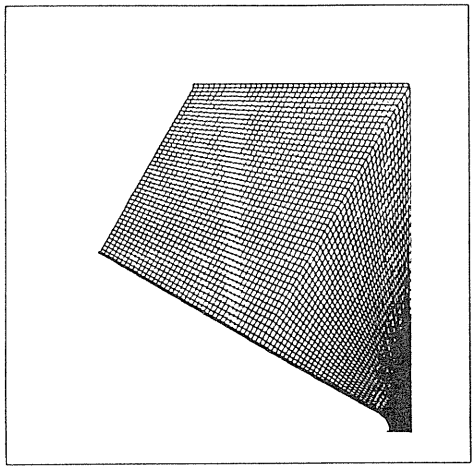


Fig. 5.10. Grid points for a blunded cone geometry. In this case, parameters jj_3 and jj_4 are equal to 0.

5.3.3. *Grid System*

The grid spacing for ξ-direction is initially equally distributed. The grid distribution can be modified during calculation using the ADPGRD subroutine although, in case of upper reflecting wall, this subroutine should not be used. The grid spacing in ξ-direction is distributed according to the ratios

$$\frac{alpha}{jj1}, \quad \frac{al1}{jj2}, \quad \frac{alpha2}{jj3}, \quad \text{and} \quad \frac{al2}{jj4}.$$

Coordinates are given in a cartesian frame system. Subroutine MTRX is in charge of converting this set of coordinates to the general $\xi - \eta$ frame system. The user can eventually modify the overall structure of the numerical grid. The keypoint is to define a set of coordinates *dgrd(j,k,1)* and *dgrd(j,k,2)* as output of subroutine GRID. Third index *1* and *2* represents the x- and y-direction, respectively. Note that modifying the sequence and the nature of the grid input-parameters can jeopardize proper use of adaptive grid. Errors can then be avoided by either switching off call to subroutine ADPGRD (*bgrd* = 'N') or modifying subroutine ADPGRD itself. Similarly, when the upper boundary of the numerical domain is a rigid wall, the use of an adaptive grid technique requires from the user to modify subroutine ADPGRD.

5.4. Convergence and Accuracy

As mentioned in section 5.1.1, the permissible timestep is limited by the maximum speed of propagation of a one-dimensional problem. For stability purpose, the *Courant–Friedrichs–Lewy* or *CFL* condition imposes the Courant number (noted *CFL*–number) to be less than unity. Theoretically, the CFL–number can be as large as 0.9, although because of the explicit scheme, values close to unity can lead to instabilities and inaccuracy. The developed non-reactive code ensures good convergence when a *CFL*–number equal to 0.5 is used. No specific problem has been reported for values up to 0.7. On the other hand, one must be careful when computing reactive flows. The *CFL*–condition used in the algorithm is based on fluiddynamic consideration and the characteristic time of the chemical source terms is not taken into account for the calculation of the timestep. Therefore, when very fast heat release occurs (high Mach numbers resulting in high temperature gradient, geometries resulting in large stagnation-zones...) the *CFL*–number must be reduced down to 0.3 or less. Numerical evidences for inappropriate *CFL*–numbers are for instance negative species densities, causing the algorithm to fail. Moreover, the convergence process also can be affected by the physical size of the numerical cells.

Section 5.1.7 describes how the code keeps tracking the convergence process. The user can use the *aercn* parameter (minimum residual) to strengthen the convergence criterion. Good convergence has always been observed when the residual becomes less than exp-05. One must be aware that computations characterized by residuals oscillating around exp-04 should not be considered as 'good converged calculations', even when the overal shock-structure remains unchanged. Note that oscillating residuals are typically observed in reactive computations.

These considerations should be used as guidelines and not as absolute rules. It is always the investigator's responsability to analyse and interpret the convergence process.

6. Nonreactive Simulations

The code has been written in a general manner so that it can be used even if the user does not have fully understanding of the code structure. Specific applications of the code are described below to show the requirements to perform a simulation and how to interpret the output. This section deals with nonreactive calculations. One must remember that output from these nonreactive simulations are readily used as initial flowfield for the reactive code. Sample calculations of reactive mixtures are described in the next chapter. Calculations with various geometries are carried out in order to show the user how to deal with the grid system and the adaptive grid technique. Detailed description of each set of input parameters is given so that inexperienced user can reproduce the computation presented. Some typical output

are presented. Graphic Software is provided by the Computer Center of the Nagoya University.

6.1. *Hypersonic Flow around a Wedge*

Steady nonreactive flow is generated around a two dimensional blunt wedge. Incoming gas is air and free stream velocity is 2100 m/s (Ma = 6.06). Initial pressure and temperature are 1.0 atm and 298 K, respectively. The complete set of numerical input is shown in Table 6.1. Note that for the present computation, parameters *ards2*, *al2*, and *alpha2* are meaningless.

Table 6.1. Input parameters for a two dimensional wedge geometry. Computation performed with these parameters is shown in Fig. 6.1.

I,ROE,1,GT,2	cans,bavt,baod,btst,lpln
0.80,0.5,20000	ddlt,dcfl,lend
0,0	linf,lfgr
N,Y	baxs,bgrd
-5.0,1.0	aercn,dcnd
1.0,3.76	anm(1),anm(2)
85, 45	jj,kk
2100.,0.,1.00,298.	dui,dub,dprs,dt
0.003,0.002,0.05,0.04,0.003	ards,ards2,al1,al2,al3
0.025,60.0,5.0	ah2,alpha,alpha2
15,70,0,0	jj1,jj2,jj3,jj4

According to the ADPGRD-algorithm the grid has been adapted 4 times. Fig. 6.1 shows successive grids: initial, after first call ADPGRD, and final grids. Pressure contours at corresponding timesteps are also drawn. Final convergence (residual < -5) is achieved after about 13,000 iterations requiring about 224 seconds CPU time on the VP2600 supercomputer at Nagoya University. The time history of the residual is shown in Fig. 6.2. Note that discontinuities occur in the residual curve each time the convergence and accuracy parameters (entropy factor and order of the TVD-scheme) are tightened up.

6.2. *Hypersonic Flow around an Axisymmetric Body*

Calculation similar to the previous one is now carried out for axisymmetry geometry (*baxs* = 'Y'). All other numerical and physical parameters (Table 6.1) remain unchanged. Fig. 6.3 shows contours af various physical quantities. The latter figure is scaled as Fig. 6.1 so that the influence of the geometry on the overall shape and angle of the bow shock can be noticed. Similarly to previous calculation, the residual history shows discontinuities when grid is adapted and convergence parameters are modified (Fig. 6.4). Convergence is reached after about 20,000 iterations, i.e. after 346 seconds CPU time.

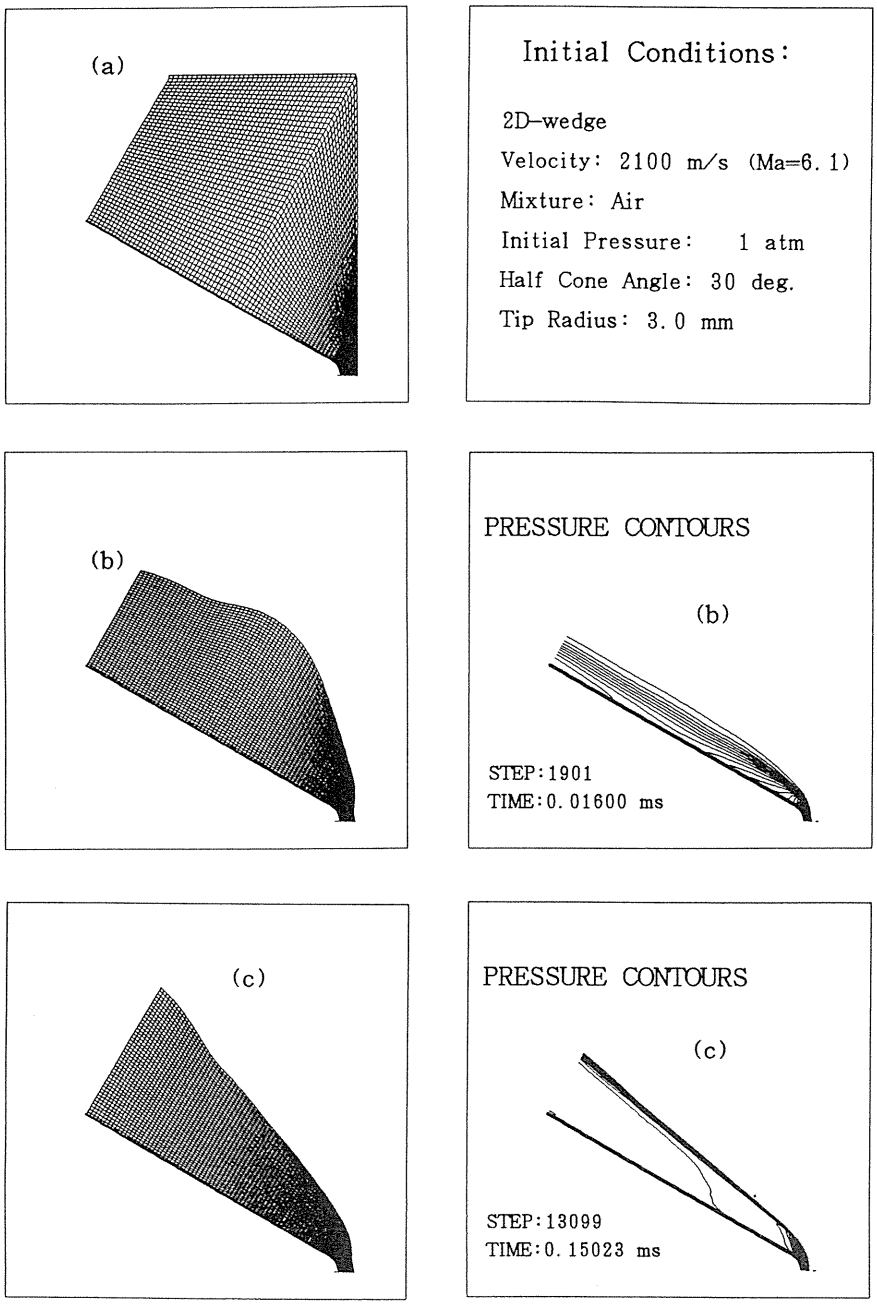


Fig. 6.1. Grid points and pressure contours of a simulation carried out with a 2D-wedge geometry at various timesteps: (a) initial (b) after first call to ADPGRD, and (c) after full convergence.

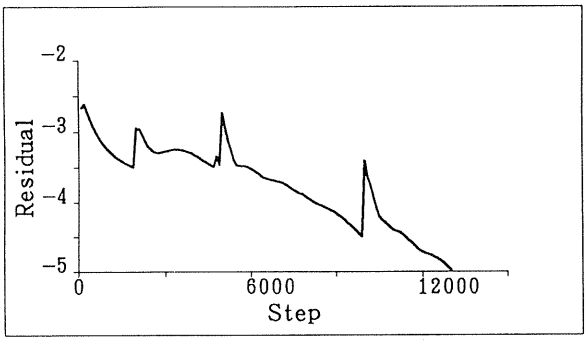


Fig. 6.2. Residual as function of timestep for simulation of a flow field around a 2D-wedge. Convergence is assumed when residual becomes smaller than −5.

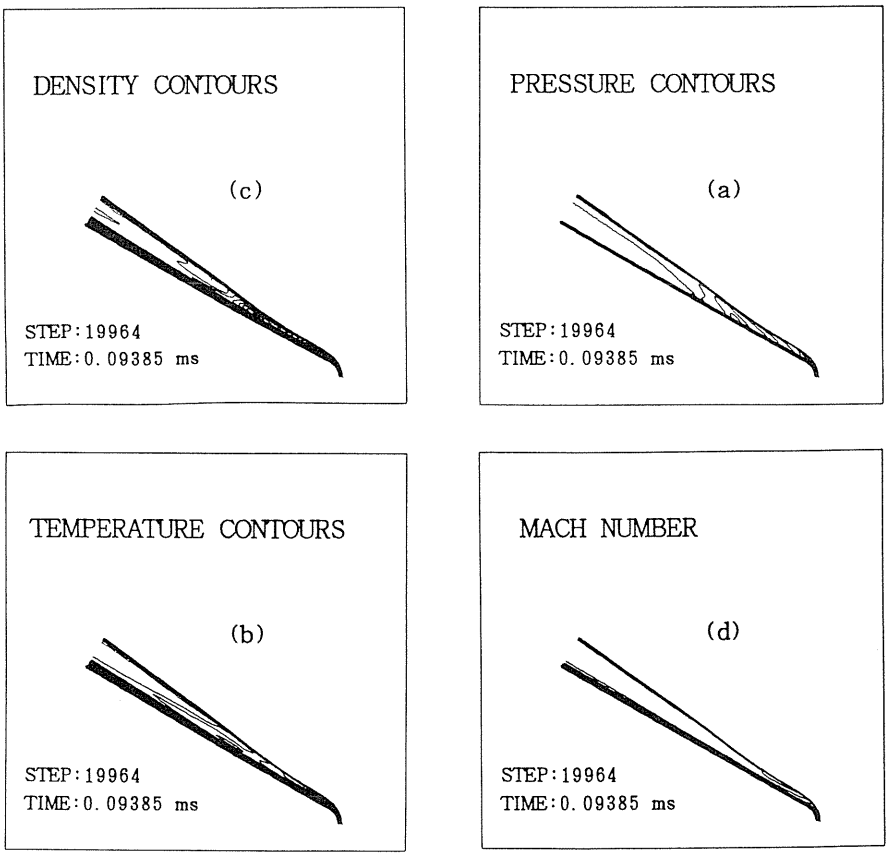


Fig. 6.3. (a) Pressure, (b) Temperature, (c) Density, and (d) Mach contours for axisymmetric computation. The solid dark line in the Mach number contours is the sonic line.

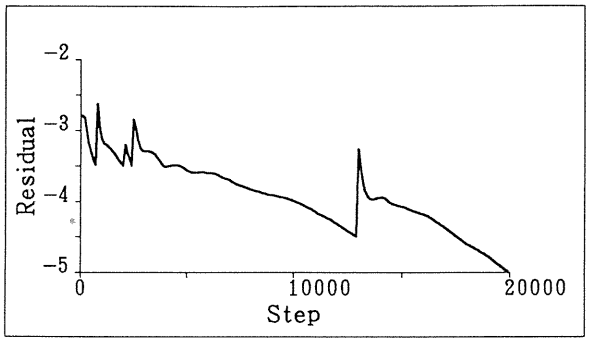


Fig. 6.4. Residual history for axisymmetric simulation.

6.3. Shock in a RAMAC Configuration

Finally a sample of nonreactive flow computation is given by simulating a flowfield in a RAMAC device. The overall numerical domain is represented in Fig. 2.1. Parameters used and physical scale are reported in Table 6.2. The geometry is axisymmetric. Subroutine ADPGRD is switched off. As indicated in section 5.1.6, boundary conditions are free slip reflection condition for the upper wall and adiabatic slip condition on the body surface. The structure of the numerical grid is shown in Fig. 6.5. Fig. 6.6 shows a selection of contours when convergence is achieved. The bow shock and the successive reflected shocks appear clearly on the pressure contours. The dark solid line in the Mach number contours is the sonic line. One can notice that the flow remains supersonic all over the domain except in a small region close to the surface of the after body. The flowfield represented in Fig. 6.6 is steady; the convergence required 21865 timesteps to be achieved. One must note that the more sophisticated the geometry is the more difficult the convergence process can be.

Table 6.2. Input parameters for a RAMAC geometry.

I,ROE,1,GT,2	cans,bavt,baod,btst,lpln
0.80,0.5,25000	ddlt,dcfl,lend
0,0	linf,lfgr
Y,N	baxs,bgrd
-5.0,-1.0	aercn,dcnd
1.0,3.76	anm(1),anm(2)
85, 45	jj,kk
2100.,0.,0.100,298.	dui,dub,dprs,dt
0.002,0.002,0.04,0.04,0.002	ards,ards2,al1,al2,al3
0.004,70.0,5.0	ah2,alpha,alpha2
10,35,5,35	jj1,jj2,jj3,jj4

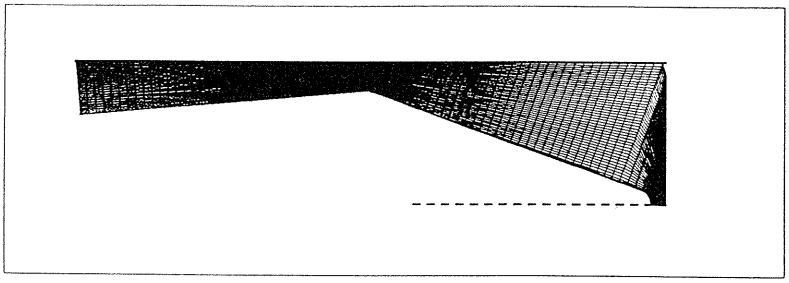
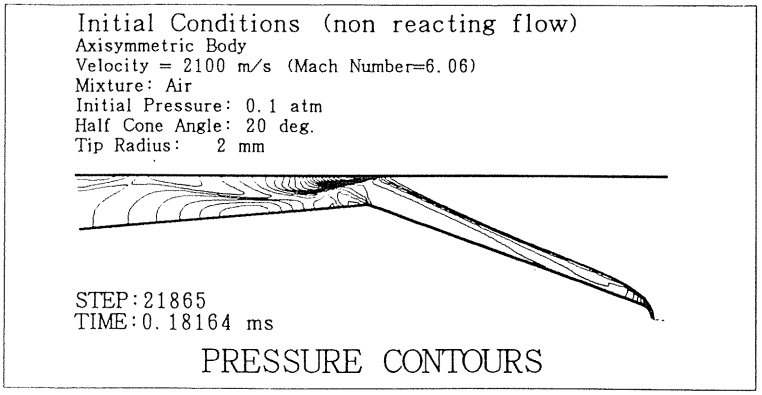


Fig. 6.5. Grid points for a RAMAC-configuration. Geometry data are given in Table 6.2.

(a)



(b)

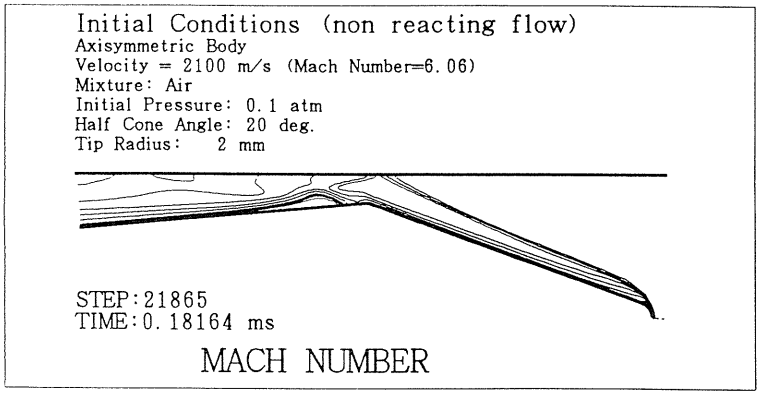


Fig. 6.6. (a) Pressure and (b) Mach contours for nonreactive flow in a RAMAC device.

The particular example shown here ensures good convergence after less than 22,000 timesteps; but convergence is quite sensitive to most input parameters: half cone angle, length of the nose, minimum free space between projectile and tube, freestream velocity and so forth. The residual curve for the present computation (Fig. 6.7) shows a quasi constant value of -4 during a long period of time. This time corresponds to the time the flow needs for being settled above the conoid aft. In others cases involving different parameters, convergence may keep oscillating between $-4.5 < \text{residual} < -4.0$. It must be clear that convergence must not be taken for granted as it is essentially dependent of the physical characteristics of the studied system. It is up to the user to analyse the convergence process on a case to case basis.

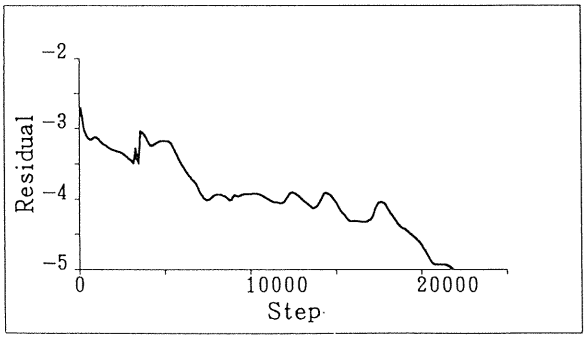


Fig. 6.7. Residual history for computation of a nonreactive RAMAC device.

7. Reactive Simulations

A series of reactive flow simulations are described in this section. Reactive flow calculation is performed with Code XXX2. Initial flow field is always provided by a preliminary nonreactive flow computation. In this section, we do not discuss further the features of these preliminary calculations. We only focus on the specifics of the reactive modeling. Meaningful results are also discussed and interpreted from a physical point of view. In the following sections, three sets of simulation are presented:

(1) Simulation of combustion processes induced by a supersonic blunt conical body. In this series of modeling, we scan a whole range of freestream velocities and nose bluntnesses in order to investigate their influence on the combustion process.

(2) Modeling of an overdriven detonation supported by a blunt cone. A strong overdriven detonation, supported by a Mach reflection configuration, is described in detail.

(3) Combustion in a RAMAC environment. The geometry and physical parameters of the sample calculation reported in 6.3 is implemented in the reactive code and simulates some features of a RAMAC device. Various inflow velocities and mixture compositions are used to describe different properties of the reactive flowfield.

7.1. Combustion Induced by a Supersonic Blunt Cone

The reactive mixture studied is a stoichiometric hydrogen-air at 10,000 Pa. The initial temperature is 298 K. The overall geometry of the conical body is shown in Fig. 7.1. The cone half angle is 30 degree and is kept constant in the present calculations. The numerical parameters used are:

ddlt:	0.15	*dcfl:*	0.5	*baxs:*	Y	*aercn:*	−5
dcnd:	1	*jj:*	85	*kk:*	55		

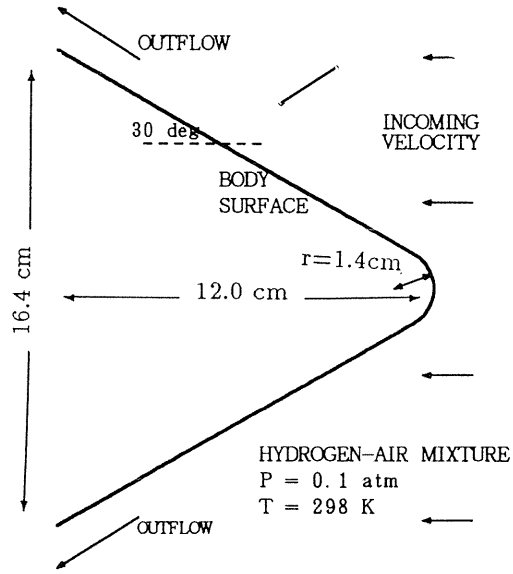


Fig. 7.1. Schematic diagram of the physical model.

Scanning a range of incoming Mach numbers and tip radii enables us to define four modes of shock-induced combustion. An overview of typical structures is shown in Fig. 7.2. The figures represent contours of the rate of H_2O production, and hence show the location of the heat release. The dashed line is the location of the front shock.

Fig. 7.2(a) is calculated for an incoming Mach number and a tip radius equal to 3.5 and 14.0 mm, respectively. Exothermic process is triggered at the tip of the body and a slow combustion is observed along the body surface. The chemical reaction zone fails to couple with the bow shock and we obtain a steady structure characterized by a shock-induced deflagration.

Fig. 7.2(b) is calculated for an incoming Mach number and a tip radius equal to 8.0 and 14.0 mm, respectively. Because of the high Mach number and of the relatively large size of the spherical nose, the chemical reaction couples readily with the whole structure of the shock and forms a steady shock-induced detonation.

Figs. 7.2(c) and (d) are the intermediate cases, both calculated with a tip radius equal to 14.0 mm. In some cases like the one in Fig. 7.2(c), we have not succeeded to converge to a

steady state, i.e. the residual never becomes less than exp-05. A strong ignition occurs at the left side of the numerical domain and a forward-running detonation wave breaks out. Fig. 7.2(c) has been calculated for Ma = 5.0, showing a snapshot of this phenomenon. The structure shown in Fig. 7.2(d) results from a Mach number equal to 5.1. The structure is steady and we observe, close to the nose, a decoupling process between the leading shock and the chemical reaction. Further, the chemical reaction succeeds to form a steady oblique detonation wave by forcing the initial oblique shock to deflect.
All the contours in Fig. 2 but (c) represent steady flowfields, i.e. the numerical simulations after full convergence. Fig. 2(d) is a part of dynamic sequence which is modeled and investigated in more details in section 7.2. We now take a look at the changes in shock structure as the Mach numbers increase, while keeping the tip radius constant. Analyses are performed for both small radius (2.25 mm) and large one (14.0 mm).

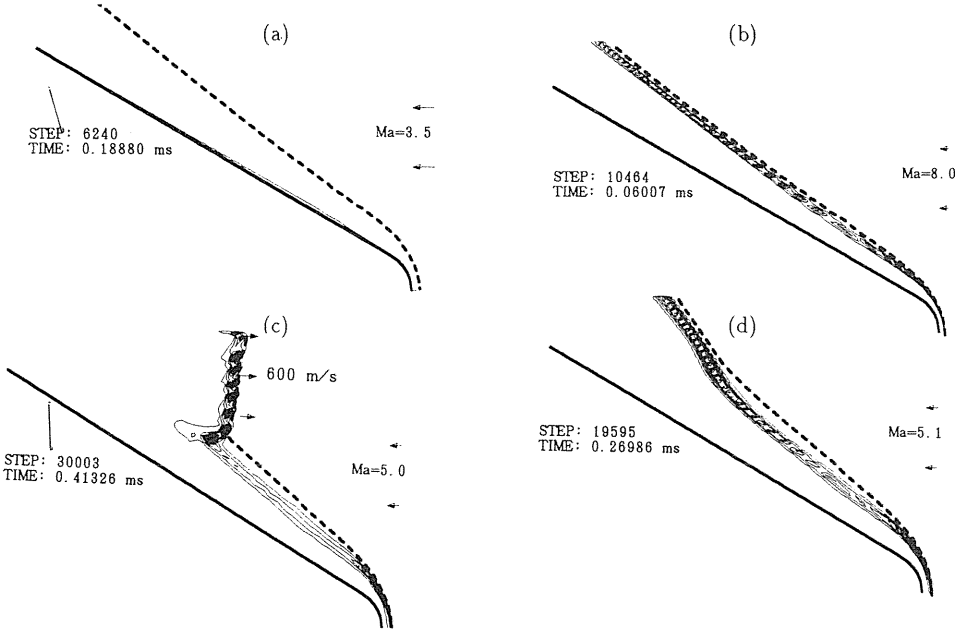


Fig. 7.2. Samples of the four combustion modes. The contours represent the production rate of H_2O. (a) Shock induced combustion, (b) Oblique detonation, (c) Runaway detonation, and (d) Shock induced deflagration developing into steady oblique detonation. The dashed line show the location of the leading shock. The normal part of the front shock in (c) is traveling forward with a velocity of 600 m/s.

7.1.1. Small Tip Radius—Increasing Mach Numbers

Fig. 7.3 shows the location of the exothermic reaction relative to the leading shock for various inflow velocities. The Mach numbers are 5.96, 7.69 and 8.43, respectively. The contours represent the density of water, while the dashed lines show the location of the leading shock front. In Fig. 7.3(a), inspite of a relatively large Mach number (6.00), the reaction zone

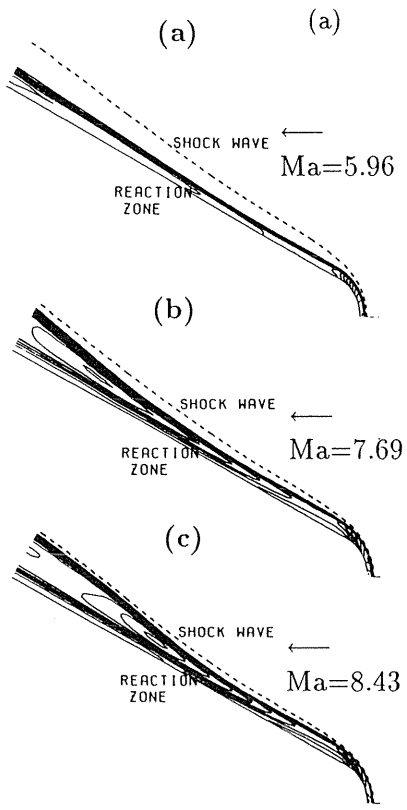


Fig. 7.3. H_2O contours of steady flow for increasing Mach numbers. The incoming Mach number is (a) 5.96, (b) 7.69, and (c) 8.43. The tip radius is equal to 2.25 mm.

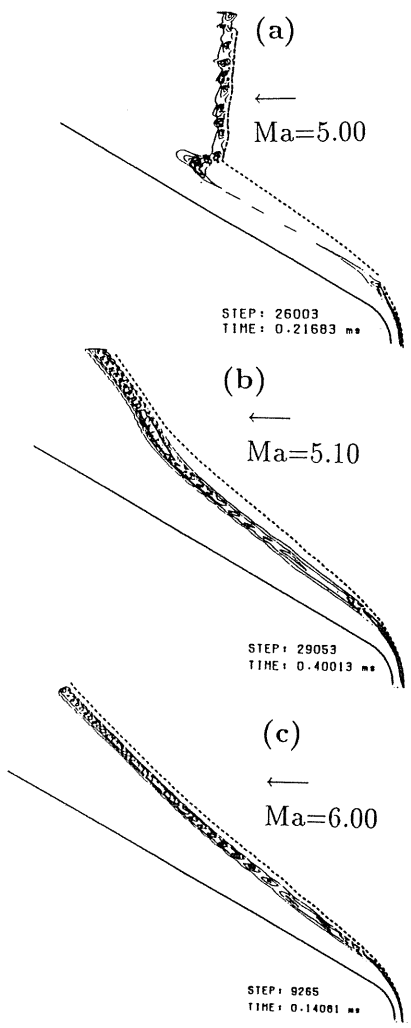


Fig. 7.4. Production rate of H_2O for increasing Mach numbers and large tip radius. The incoming Mach number is (a) 5.0, (b) 5.1, and (c) 6.0. The tip radius is equal to 14.0 mm.

does not couple with the leading shock. In this case, the Mach number normal to the shock at the end of the computational domain is equal to 3.7, which is much lower than the CJ-value. As the incoming flow velocity increases, we gradually observe a coupling between the shock and the reaction zone: this occurs for the first time for an incoming Mach number equal to 7.44. The Mach number, normal to the shock where coupling occurs, is then equal to 4.7, close to the CJ-value. For incoming Mach numbers higher than 7.44, the shock/reaction coupling occurs closer to the tip, as seen in Figs. 7.3(b) and (c). It is interesting to notice that

the coupling process occurs smoothly without drastic and sudden increase of the angle of the standing detonation, in contrast to simulations carried out on a sharp wedge at a high Mach number, as reported by Li[16]. One can notice that the standoff distance (SD) evolves little with increasing Mach numbers. In all these calculations, the SD ranges from 0.393 mm to 0.301 mm, for Mach number going from 4.4 to 8.4. The overall shape of the leading shock does not vary much between the extreme cases. The deflection angle between the body surface and the leading shock increases slightly from 8 degrees (Ma = 6) to 10 degrees (Ma = 8.4). Moreover, the steady structures show little deflection relative to the cone angle, compared to the previous numerical studies carried out on 2D wedges[34].

7.1.2. Large Tip Radius—Increasing Mach Numbers

When the tip radius is large, the simulations carried out with extreme Mach numbers show similarity with those described above (Fig. 7.4). At low incoming velocities, the temperature behind the bow shock is low and the induction length is too long to generate combustion processes near the tip. Some exothermic reactions occur over the body surface. At relatively high Mach numbers (higher than 6), a full coupling between the chemical reaction and leading shock is observed and a steady shock-induced detonation is established (Fig. 7.4(c)). The transition between these two extreme structures occurs for a range of Mach numbers, where no steady flows can settle. As the Mach number increases, the reaction zone comes closer to the leading shock and an increasing amount of energy is released behind it. At a critical Mach number, enough energy is released to trigger a detonation front, which the incoming flow is not able to stabilize. A normal detonation have starting far from the nose appears, moves forward, and passes by the solid body (Fig. 7.4(a)). When the Mach number increases further, the reaction zone keeps moving close to the leading shock and steepens the original oblique shock. But the latter is now strong enough to be stabilized and to form a steady oblique detonation front (Fig. 7.4(b)).

The coupling process between the shock and the heat-release zone occurs in that case quite smoothly: away from the nose, and after a short decoupling of the combined *oblique shock/chemical reaction.* Contrasting with the calculations performed with small tip radius, we observe a deflection of the leading shock. This is similar to the one reported by Li[16].

The transition from a strong forward-running detonation (Fig. 7.4(a)) to a steady detonation (Fig. 7.4(b)) is very sentitive to small changes in the incoming flow velocity: Figs. 7.4(a) and 7.4(b) are calculated for Mach numbers equal to 5.0 and 5.1, respectively. Notice that both Figs. 7.4(b) and 7.4(c) show contours of steady flow, i.e. a converged numerical simulation; on the other hand, Fig. 7.4(a) shows only a snapshot of a non-stationary flow.

7.1.3. Overall View

Fig. 7.5 gives an overall view of all the calculated structures. The marks refer to the four aforementioned types of shock structures that we observed in our simulations. The whole domain of the *radius-size/Mach number* plane is divided into

(1) an upper part where steady shock-induced oblique detonation waves are recorded,

(2) a lower section where steady shock inducing combustion processes over the body surface are observed,

(3) a transition zone whose characteristcs depend on the size of the nose.

When the tip radius is small, a limited range of relatively high Mach number gives the transition from the deflagrative mode to the detonative one. Because of the high Mach numbers at which the transition occurs, the process is smooth and the shock structure does not evolve into a non stationary mode.

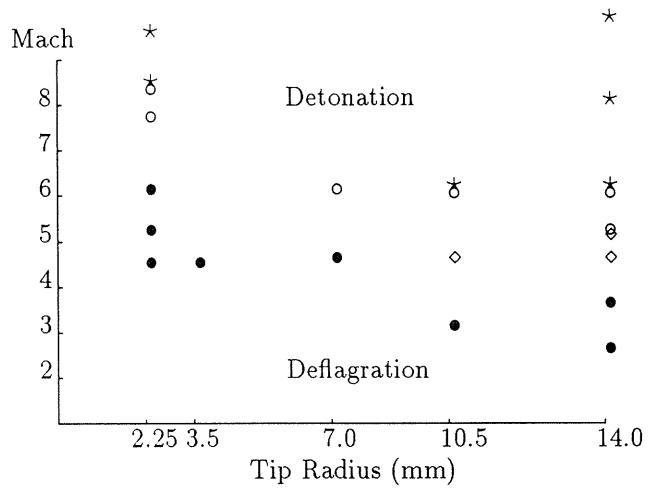


Fig. 7.5. Overall distribution of calculated reaction-shock structures as function of tip radius and Mach number. (•) no coupling (steady shock induced combustion), (○) steady coupling away from the nose, (⋆) full coupling (steady shock induced detonation), (◇) forward-traveling detonation.

As the tip radius increases and, consequently, the Mach number causing the transition lowers, a two-stage transition appears: (1) At low Mach numbers, an unsteady detonation develops, while (2) at high Mach numbers, a steady detonation is formed, away from the nose, being characterized by the deflection of the bow shock. For large tip radii (14.0 mm), relatively low Mach numbers (from 3.5) are able to trigger the forward-traveling detonation structure.

Note that the runaway normal detonation in Fig. 7.2(c) is the so called *unstart* phenomenon observed in ram-accelerator experiments, of which the origin is still unknown. In the next section, we try to get better understanding of the physical properties of this overdriven normal detonation wave.

7.2. Overdriven Detonation Supported by a Blunt Body

7.2.1. Transition from Deflagration to Detonation

Sequences and particular structures of the forward-running strong detonation are shown in Figs. 7.6 and 7.7. The former represents pressure contours and the latter shows the Mach number. The sonic line is the black solid line and the dashed line in front of the Mach contours shows the location of the front shock. The detonation has already been ignited at the downstream base of the cone and the snapshots show the wave on its way to pass over the cone. As the chemical reaction develops behind the initial bow shock, the angle of the bow shock at the left side of the computational domain increases first and allows a triple point to form. One can notice the strong triple point and Mach stem close to the cone surface in Fig. 7.6(a). At that point, the leading shock gains tremendous strength, becomes almost normal to the incoming velocity and starts to move forwards. A large Mach-reflection configuration is formed and supports the normal overdriven detonation (Fig. 7.6(b)). The velocity of the normal detonation, relative to the freestream velocity is almost constant throughout the normal

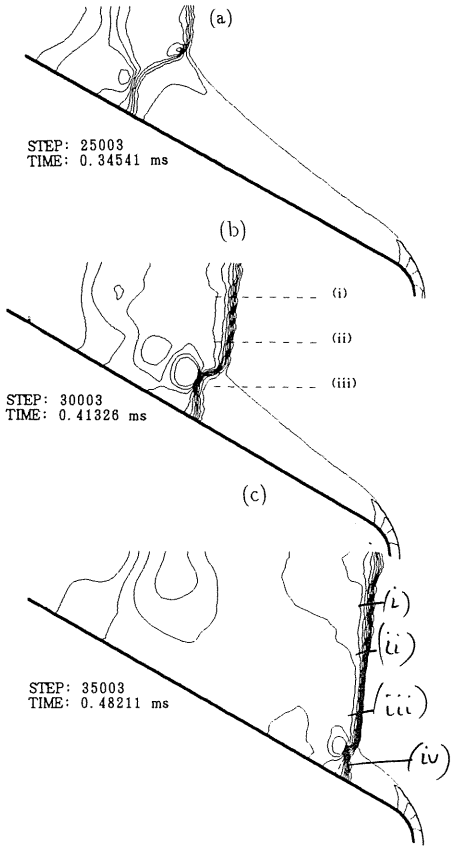


Fig. 7.6. Sequence of pressure contours when a strong unsteady detonation is passing by the projectile. The incoming Mach number is 5.0.

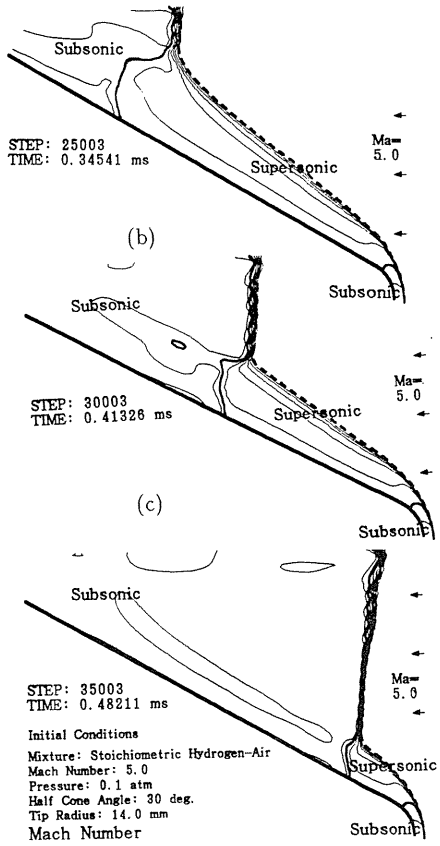


Fig. 7.7. Sequence of Mach number contours when a strong unsteady detonation is passing by the projectile. The incoming Mach number is 5.0.

part of the shock front and is approximately equal to 600 m/s as shown in Fig. 7.8. This figure shows the relative velocity of the front shock at 4 different heights (i), (ii), (iii), and (iv) above the body surface (see Fig. 7.6(c) for the definition of the locations). When the initial leading shock is not yet driven by the forward-running normal detonation, its position is quite stable (velocity in excess = 0.0 m/s). As soon as the normal detonation passes by, an abrupt motion of the front shock is noticed and its velocity stabilizes quickly around 600 m/s. Behind the normal and reflected shocks, the flow is subsonic as shown in Fig. 7.7, accounting for the high pressure and density in this domain. The cause of the strong detonation which is in a way stabilized in a super-CJ incoming flow can be attributed to the pressure field behind the triple shock intersection. The pressure rise is much higher than P_{CJ} because of its hypersonic Newtonian-flow character. Since the flowfield behind the detonation is subsonic, the pressure field is nearly uniform, enabling the normal strong detonation to exist through the proper Rankine-Hugoniot relations.

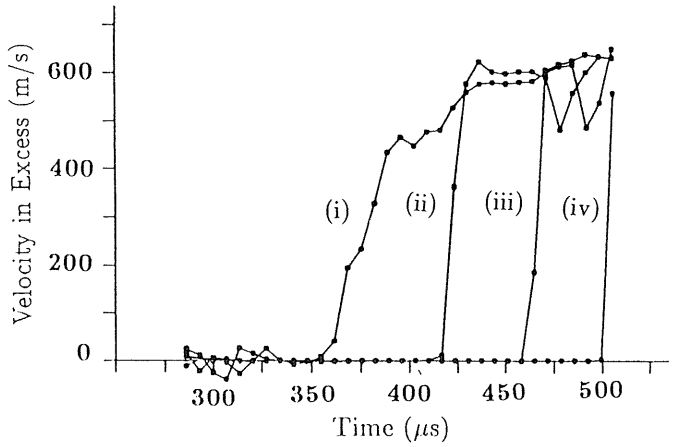


Fig. 7.8. Velocity in excess of the normal runaway detonation. The labels (i), (ii), (iii), and (iv) refer to their location in Fig. 7.6(c).

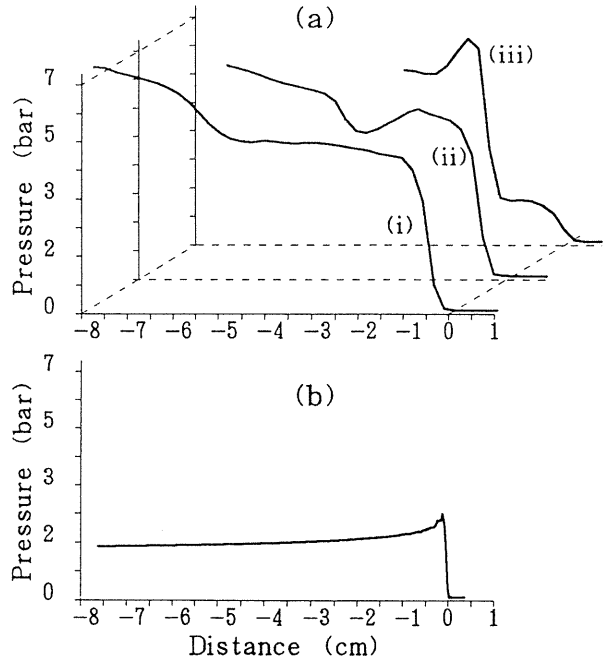


Fig. 7.9. Pressure profiles: (a) From the forward-running overdriven detonation shown in Fig. 7.6(b); labels refer to the locations where the profiles are recorded. (b) From a 1D-computation. The origin of the absissa is the position of the leading shock.

Pressure profiles have been recorded at different heights (i), (ii), (iii), above the cone. The locations of the *numerical probes* are shown in Fig. 7.6(b). Fig. 7.9(a) represents the three pressure curves as functions of distance behind the shock. One can notice the high pressure jump (4 to 5 bar, i.e. a pressure ratio of about 40 to 50) behind the normal shock. Moreover, the cone-ramp prevents the pressure to drop and to expand behind the reaction zone. For comparison purposes, Fig. 7.9(b) gives the pressure profile computed using a one-dimensional detonation model for the same mixture and the same upstream pressure and temperature. In that case the 1D planar detonation wave is traveling at CJ-velocity. The von Neumann spike for this 'regular' CJ detonation is about 2.5 bar, i.e. half the value observed in Fig. 7.9(a), and is followed by the Taylor expansion.

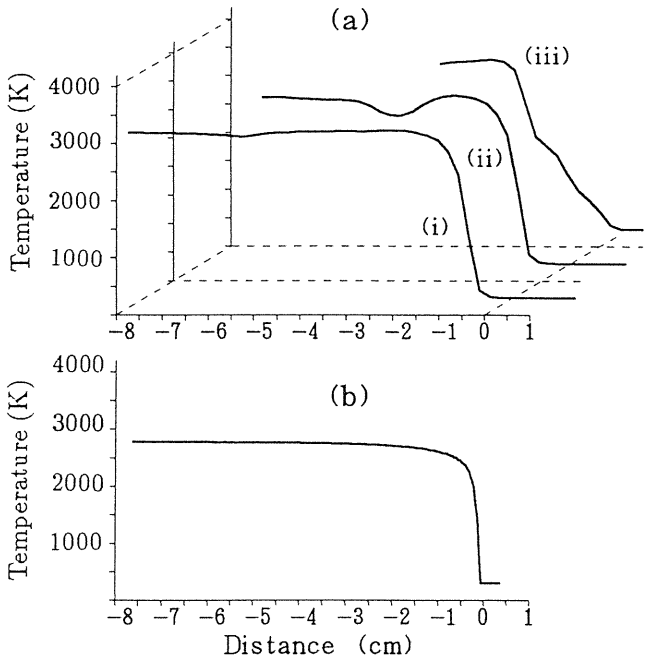


Fig. 7.10. Temperature profiles; conditions are identical to Fig. 7.9.

On the other hand, similar comparison between the CJ-temperature profile and the temperature profiles recorded from the flowfield 7.6(c) does not reveal considerable discrepancies (Fig. 7.10). The velocity of the unsteady normal shock is much higher than the CJ-velocity, accounting for the higher maximum temperature observed in Fig. 7.10(a).

Although the forward-traveling detonation is normal to the incoming flow and is quasi planar, its charateristics are clearly influenced by the presence of the conical projectile and do not show any similarities with CJ plane detonation. On one hand, the chemistry-dependent characteristics are not strongly affected, because the temperature profiles are similar. On the other hand, the dynamic behavior of the detonation wave is deeply modified by the cone. The strong downstream pressure, stemmed from the presence of the conical projectile, sustains the normal strong detonation and accelerates it to a velocity much higher than the CJ

one. This proves to be the main influence of the downstream field on the detonation characteristics.

The Chapman-Jouguet detonation is only a particular case where the unsteady downstream flow fits well into a steady leading shock plus chemical reaction. Quasi CJ-conditions occur experimentally in simple obstacle-free geometries. Nevertheless, using more complicated geometries, one may expect quasi CJ-conditions not to be fulfilled, as illustrated by the present work. This is of most importance for the application of oblique detonations, i.e. for a RAM-accelerator device. Accordingly, shock structures and features have to be studied more carefully.

7.2.2. *Stability of oblique detonations*

We now investigate how stable the oblique detonations described in Figs. 7.2(b) and (c) are. We started the computation using a fully converged solution (Fig. 7.11(a)) of a reactive flowfield characterized by the incoming Mach 8.0 (3250 m/s). The bow shock is perfectly steady and we observe a full coupling between the shock and chemical reaction. The freestream velocity is then gradually lowered. As the flow decelerates via different velocity regimes down to Ma = 5.0, the combustion process changes accordingly. And subsequent accelerations up to the initial high Mach number force the overall structure of *shock-reaction interaction* to regain its initial shape, i.e. the one shown in Fig. 7.11(a). We can conclude from this simulation that the oblique detonation shows some stability. Moreover, if the freestream Mach number shortly decreases down to a value lower than Ma = 5.0, allowing the chemical reaction to trigger a triple point and to generate a Mach reflection, similar to the one shown in Fig. 7.6, subsequent accelerations up to the original Mach number (Ma = 8) are unable to bring the run-away normal detonation back to the initial oblique detonation. The detonation is now supported by a triple point and has been strengthened (Fig. 7.11(b)).

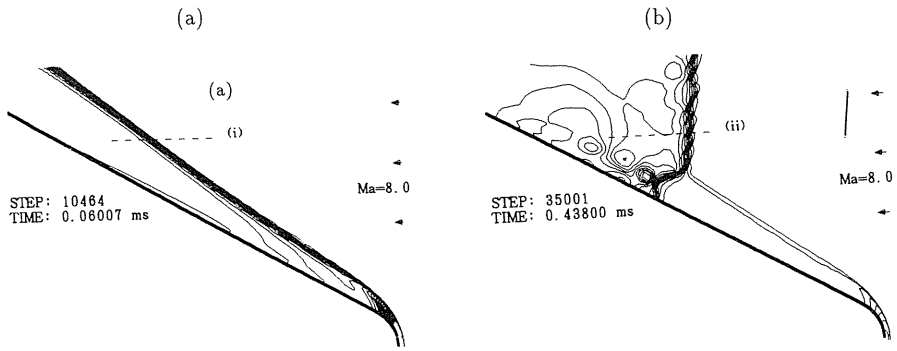


Fig. 7.11. Pressure contours for an identical freestream velocity; (a) steady flowfield, (b) overdriven forward-traveling detonation, supported by a strong Mach reflection.

The velocity must exceed Ma = 9.0 to be able to stabilize the overdriven normal leading shock, i.e. a Mach number close to twice the CJ-one. It is interesting to note that both Figs. 7.11(a) and (b) are characterized by the same freestream velocity, upstream pressure and temperature; different is only the time history of the flowfield (hysteresis effect). Curves (i) and (ii) in Fig. 7.12 show pressure profiles at an identical height above the body surface (the

location is shown in Fig.7.11) for the steady and unsteady detonations, respectively. When an unsteady normal shock has been ignited due to the presence of the cone, very high pressure is generated, leading to a strong detonation wave. The normal detonation wave gains tremendous strength by being supported by the high pressure resulting from the complex Mach reflection configuration and the high incoming Mach number. Conclusions of this simulation are similar to those discussed in the previous section, i.e. the downstream field plays a major role in determining the shock structure and imposing dynamic charateristics to the shock front. Moreover, the simulated case shows the importance and influence of the flow history on the asymptotic solution, i.e. hysteresis effect.

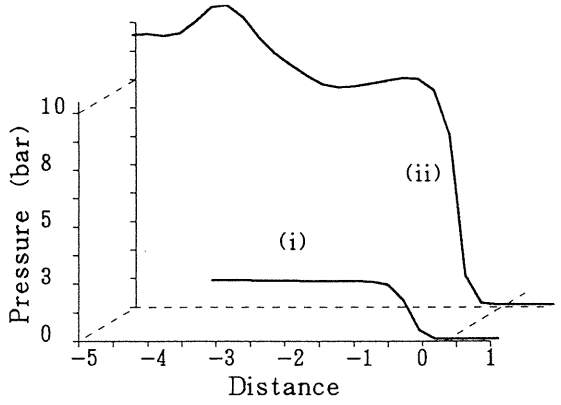


Fig. 7.12. Pressure profiles at different locations shown in Figs. 7.11. The origin of the absissa is the position of the leading shock.

7.3. Combustion in a RAMAC device

Numerical simulations of ram accelerators have been performed by many researchers and most of the investigations focus on the thermally choked and superdetonative modes[6,7,35-37]. So far, no study has been extensively carried out on the so called *unstart* phenomena. An *unstart* is defined as the 'phenomenon whereby a normal shock is disgorged from the throat and outruns the projectile[38].' In the previous section we have conducted comprehensive study of the combustion process and flow structure around a supersonic conical body. Phenomena very similar to the *unstart* have been observed. However, the technological issues of ram accelerators and ramjets have an outer boundary and a downstream throat condition as essential features. These features could change the conditions for the formation and propagation of normal detonation wave in such a way that the interpretation of Figs. 7.6 and 7.7 relative to these devices can be questionable. In the present section, we perform series of simulations using a solid cylindrical confinement around the supersonic projectile. The geometry is similar to the one described in Table 6.2 (Fig. 6.5), resulting in a diameter of the inner tube of 3.96 cm. The largest diameter of the projectile is equal to 3.16 cm. The total length of the projectile is about 8.0 cm. First, we report about calculations with no reacting mixture and conditions to achieve a steady superdetonative mode. Then we change slightly the freestream velocity and the composition in order to observe the *unstart* event.

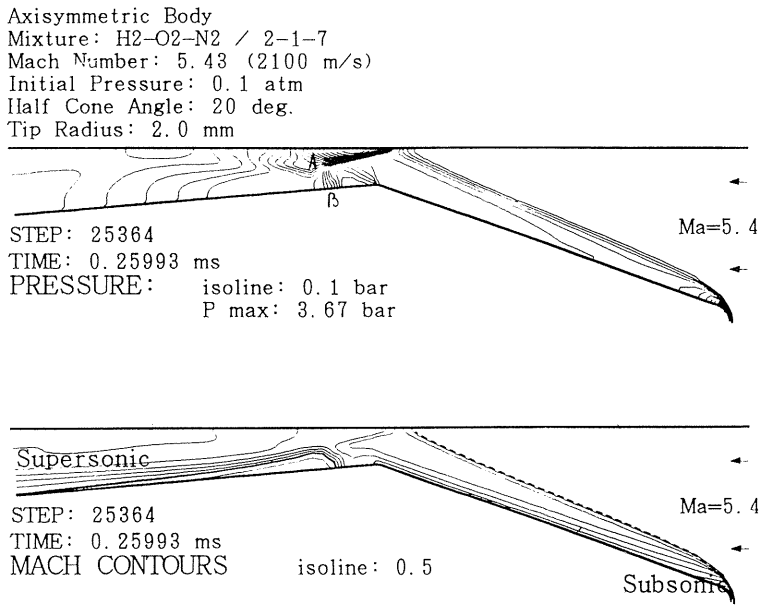


Fig. 7.13. Pressure and Mach countours when the projectile is steadily traveling at Mach 5.4. The non-reacting mixture is stoichiometric H_2-O_2, diluted with 70% N_2. The solid dark line in the Mach-contour is the sonic line.

7.3.1. Non Reactive and Reactive Calculation — Superdetonative

Fig. 7.13 shows the pressure and Mach contours for a simulation carried out in a stoichiometric hydrogen—oxygen mixture diluted with 70% of nitrogen at an initial pressure of 10,000 Pa. The inflow velocity, corresponding to the projectile velocity is equal to 2100 m/s, i.e. Mach number equal to 5.4. The chemical reaction has been turned off and the simulation shown is the result of a converged computation; the residual after 25364 timesteps becomes less than −5.0. As seen in the Mach contour, the flow remains supersonic except behind the shock AB (Fig. 7.13), normal to the surface of the projectile. When convergence is achieved, the chemical reaction is numerically turned on and computation is performed up to a new steady flowfield. Fig. 7.14 shows the flow properties when the residual of the reactive calculation is less than −5.0, i.e. the flowfield can reasonably be assumed steady. The shock structure has not changed drastically from the non reacting calculation and the high temperature behind the normal shock AB (Fig. 7.14) accounts for some chemical reaction rate. For the given RAMAC geometry and the implemented initial conditions, the chemical reaction fails to develop in the divergent section of the RAMAC device. Note that the numerical domain we used for this computation does not allow the simulation of modes like *thermal choking* (in that case, the heat release would located behind the projectile). In order to increase the reactivity of the mixture and to promote chemical reactions in the diffuser, one can change the mixture composition and/or the freestream Mach number:

- Decreasing the dilution (nitrogen content) enables us to promote the chemical reaction in the area close to the throat, although one must be carefull not to ignite chemical reactions at the tip of the projectile.

- Increasing the inflow velocity results in rising the temperature behind the shocks which enlarges the heat release zone. However high velocities can sweep away the reacting mixture resulting in the failure of the chemical reaction to stabilize in the superdetonative mode of the RAMAC.

One example of steady chemical reaction close to the throat is shown in Fig. 7.15. In that case, the reactive mixture is stoichiometric hydrogen-oxygen diluted with 57.1% of nitrogen and the freesteam velocity is equal to 2600 m/s (Ma=6.42). The chemical reaction is on the verge of going through the throat and being disgorged in front of the nose of the projectile. The steadiness of the flowfield in Fig. 7.15. has been checked by computing up to a *residual* equal to −7.

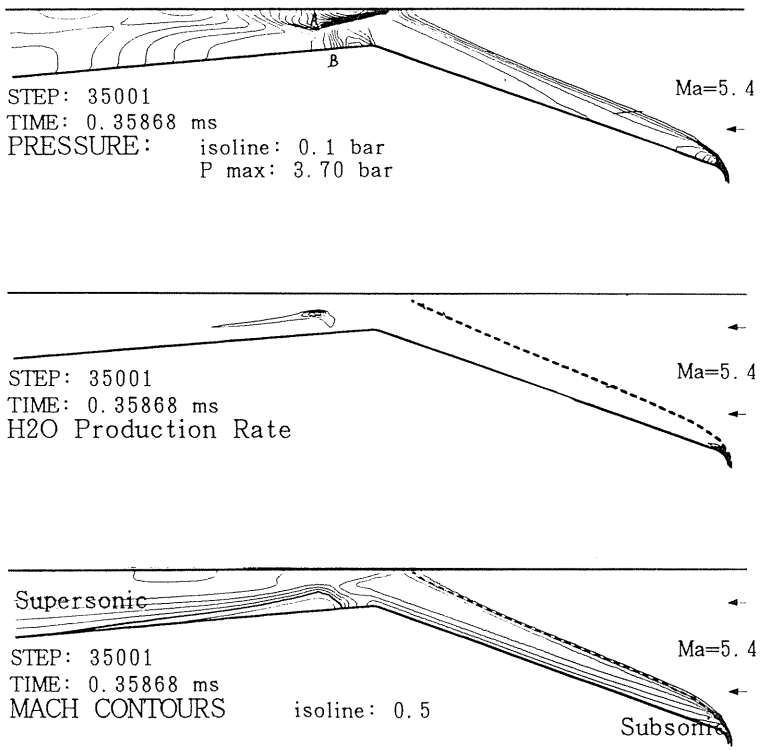


Fig. 7.14. Pressure, water production rate and Mach contours. Physical conditions are similar to these in Fig. 13. The chemical reaction has been turned on.

7.3.2. Transition to Unstart—Phenomenon

A new set of computations have been carried out to observe *unstart* in the RAMAC device. The mixture is stoichiometric hydrogen-oxygen diluted with 50% of nitrogen. The inflow velocity is 2100 m/s (Ma=5.43). Figs. 7.16 to 7.19 give a sequence of the developing chemical reaction. The chemical reaction initially ignites behind the reflected shock (Fig. 7.16), similarly to Fig. 7.14. Compared to the latter, the nitogen content is low (50% compared to 70%) and accounts for a larger chemical reactivity of the mixture. As can be seen

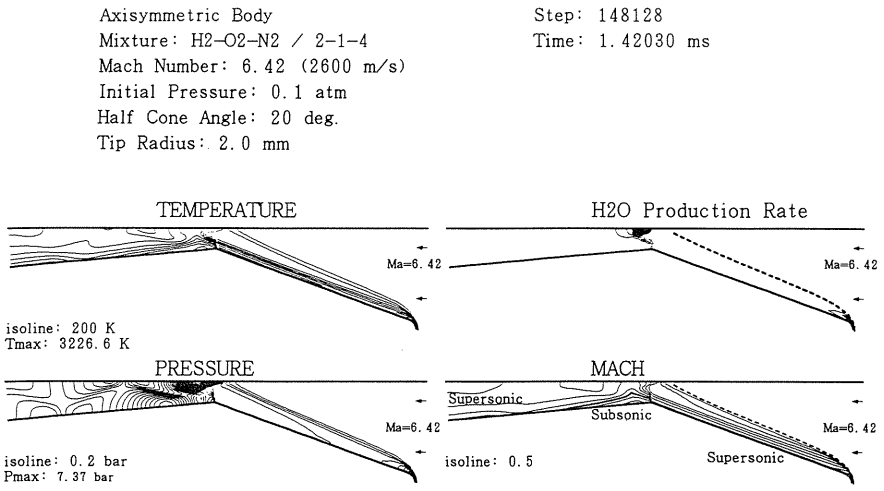


Fig. 7.15. Contours for a reacting mixture H_2-O_2-N_2/2-1-4 at an initial pressure of 10,000 Pa. The projectile velocity is equal to 2600 m/s (Ma=6.42). The chemical reaction is stabilized close to the throat of the device. *Residual* for the computation is equal to −7.

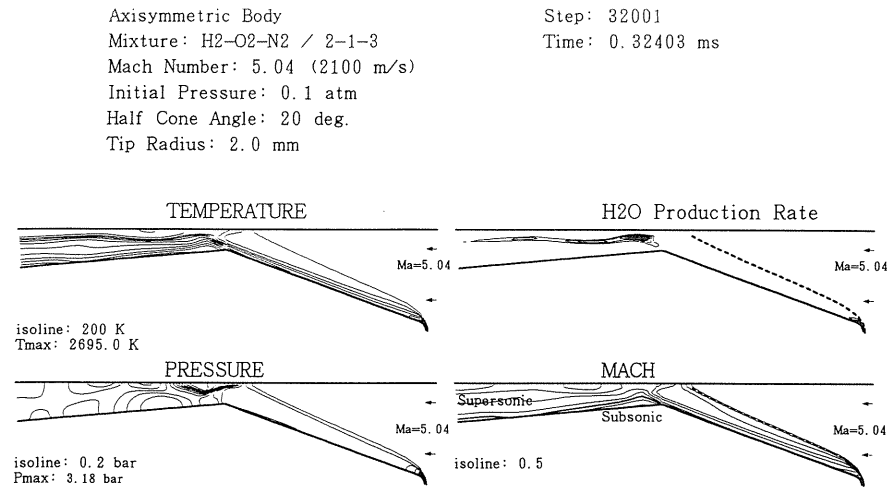


Fig. 7.16. Contours for a reacting mixture H_2-O_2-N_2/2-1-3 at an initial pressure of 10,000 Pa. The projectile velocity is equal to 2100 m/s (Ma=5.04). The chemical reaction develops behind the reflected shock, downstream to the throat.

from the water production-rate contours (Fig. 7.16) the reaction rate is intense and the heat release does not stabilize in the divergent. Note that some combustion processes are induced at the tip of the projectile nose. The heat release zone spreads then over the entire free space between wall and projectile (Fig. 7.17). The snapshot of the flowfield in Fig. 7.17 is similar

to Fig.7.15, although in Fig 7.17, the reaction zone is moving forward, i.e. the calculation does not converge: the *residual* oscillates between −2.0 and −3.0. Figs. 7.18 and 7.19 show further sequences and one can observe how the reaction zone develops in front of the conical body. When the exothermic reaction zone passes through the throat of the RAMAC the maximum pressure increases tremendeously: from 3.18 bar (Fig. 7.16) to 6.06 bar (Fig. 7.17) and 11.32 bar (Fig. 7.18). As the detonation front outruns the projectile the maximum pressure drops gradually (5.63 bar in Fig. 7.19).

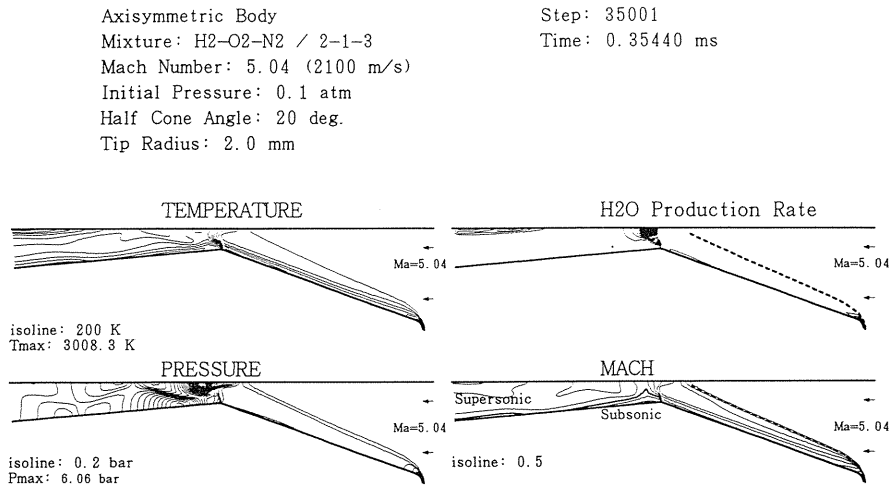


Fig. 7.17. Continued from Fig. 7.16 The chemical reaction spreads out all over the space between the projectile and the tube.

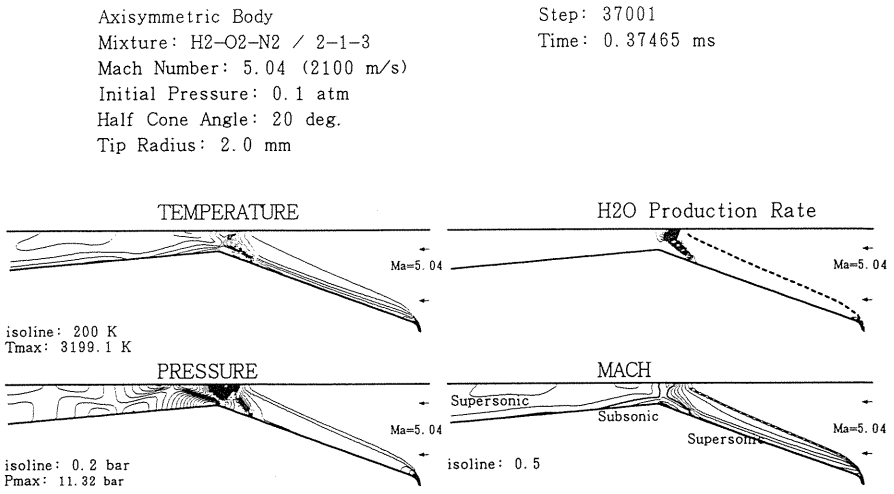


Fig. 7.18. Continued from Fig. 7.17. The chemical reaction passes through the throat. *Unstart* occurs.

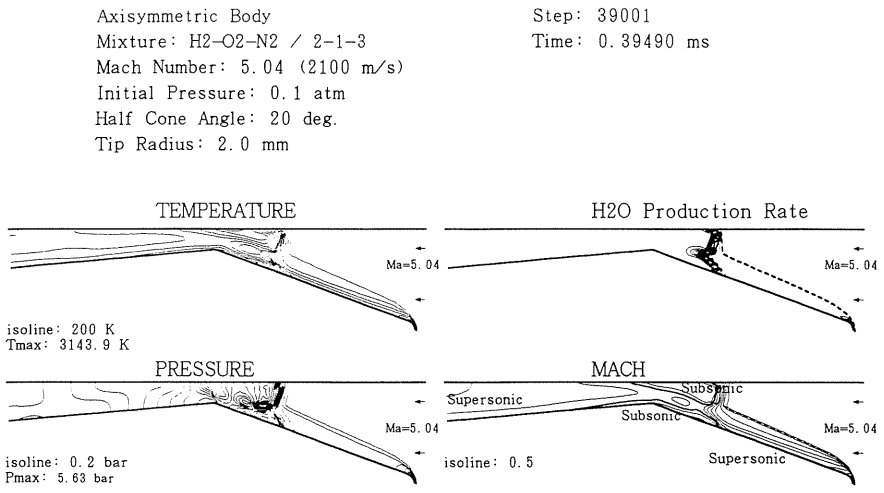


Fig. 7.19. Continued from Fig. 7.18. A strong runaway detonation develops and overtakes the projectile.

The running away detonation exhibits a normal detonation front supported by a Mach stem structure close to the surface of the cone. This shock configuration is similar to the one developed by the forward-traveling detonation described and extensively discussed in section 7.2.

8. Conclusion

This paper has described a series of codes which constitutes a robust tool for performing simulation of super- and hypersonic flows. The codes, as they stand (as of May 1994), treat a wide range of geometries and grid configurations. The sample calculations described here are bench marks and should enable all potential users to master both the reactive and nonreactive codes. With minimum changes in appropriate subroutines, large ranges of boundary conditions and source terms could be handled. For the time being, the reactive mixture is limited to hydrogen–oxygen–nitrogen but no restriction on the species ratio is imposed. We would like to point out that while the codes have been checked out carefully, by no means has every possible application been tested. Thus there may be bugs appearing in new applications. Be alert as sometimes the codes will not blow up but the solution will simply lose its physical consistency and becomes physically meaningless.

The reactive calculations performed in section 7 constitute a detailed study of oblique detonations supported by a supersonic body. It has been shown that most of the shock–combustion couplings are characterized by steady–state configurations resulting either in a induced deflagration or in a detonation wave. When the tip radius of the body is small, the coupling between the shock and the chemical reaction zone occurs smoothly and at relatively large Mach numbers. We do not observe considerable deflections of the leading shock even when the chemical reaction becomes coupled. When the tip radius is large, the transition

from an induced deflagration to a detonation does not occur as smoothly as when the tip radius is small. In some cases, we observe outrunning overdriven detonation. The transition from a steady deflagration wave to a proceeding wave occurs abruptly and is very sensitive to Mach number variations. The features of the overdriven quasi–normal detonation wave are completely different from that of CJ plane detonations and are strongly influenced by the downstream flowfield. This study shows that the characteristics of detonations around a given geometry, for example a RAMAC projectile, can be far from the CJ–prediction, and consequently one must be careful when extrapolating CJ–calculations to a detonation traveling in non–planar environments. Stability tests suggest us that, in certain cases, the shock structure is somehow dependent on the time history of the flowfield.

The simulations carried out using a ramac-geometry require deeper analyses. Nevertheless, they show that the relatively fundamental investigation on the reactive flowfield around the blunt cone is useful in terms of ramac–technology.

The presented codes will be helpful for further fundamental and technical investigations on supersonic inert or reactive flows around solid bodies and various types of projectiles. Potential applications range from safety problems in confined rooms to simulation of propulsive devices like ram-accelerators.

9. Acknowledgements

We would like to acknowledge the work and contribution of Matsuo Akiko who originally developed the main algorithm presented in this paper. The authors are grateful to the Computation Center of Nagoya University for the allowance of tens of hours of computer-time. All computations presented in this work have been carried on Fujitsu supercomputer VP-2600. The stay of M.H. Lefebvre in the Department of Aerospace Engineering has been supported by the European Union through its Science and Technology Program.

10. Nomenclature

a: Speed of sound (m/s)
A: Frequency factor
C: Chemical concentration (mol/m^3, mol/cm^3)
C_v: Heat capacity of the total mixture (J/K kg)
$c_{v,i}$: Molar heat capacity of species i (J/K mol)
CFL: Courant number
D: Time–rescaling matrix
E: Activation energy (J/mol K, cal/mol K)
$\hat{E}$: Vector of fluxes in ξ coordinates
$\hat{F}$: Vector of fluxes in η coordinates
$\hat{H}$: Vector of axisymmetric terms
Δh_f^o: Standard enthalpy of formation (J/mol)
J: Jacobian of coordinate transformation
K_{eq}^o: Chemical equilibrium constant at standard condition
k_{back}: Chemical reaction rate constant for backward reaction
k_{for}: Chemical reaction rate constant for forward reaction

Ma: Mach number
n: Temperature exponent in chemical rate constant
n_i: Number of mol by unit mass (mol/kg)
N: Number of chemical species
P: Pressure (Pa, bar)
$\hat{Q}$: Vector of conserved variables in physical coordinates
R: Universal gas constant (8.3143 J/mol K)
R_x: Transformation matrix that diagonalizes the Jacobian of $\hat{E}$ or $\hat{F}$ with respect to $\hat{Q}$ (Section 4.2)
s^o: Standard absolute entropy (J/mol K)
$\hat{S}$: Vector of chemical source terms
T: Temperature (K)
U: Total internal energy per unit mass (J/kg)
u_i: Specific internal energy of species i (J/kg)
$\mathrm{u}^o_{f,i}$: Internal energy of formation of species i (J/mol)
U: Contravariant velocity in ξ-direction
u: Velocity component in x-direction
V: Contravariant velocity in η-direction
v: Velocity component in y-direction
W: Molecular mass
α: Global third body efficiency (section 3.2)
$\alpha_{j,k}$: Third body efficiency of species j for reaction k (section 3.2)
ν: Stoichiometric coefficient
$\xi_{j,k}$: Partial order of species j for the kth forward chemical reaction
$\xi^*_{j,k}$: Partial order of species j for the kth backward chemical reaction
ρ: Density ($\mathrm{kg/m^3}$)
ω: Rate of density change due to chemical reactions
$\Delta\nu_k$: Difference in stoichiometric coefficients for kth reaction

subscript

i: Refer to species i; Grid index for ξ direction
j: Refer to species j; Grid index for η direction
k: Refer to the kth chemical reaction

superscript

n: Current iteration level

11. References

1) Chernyi, G.G., Supersonic Flow around Bodies with Detonation and Deflagration Front, Astronautica Acta, Vol.13, pp.464–480, 1967.
2) Lher, H.F., Experiments on Shock-Induced Combustion, Astronautica Acta, Vol.17, 1972, pp.589–597.
3) Hertzberg, A., Bruckner, A.P., and Bogdanoff, D.W., Ram Accelerator: A New Chemical Method for Accelerating Projectiles to Ultrahigh Velocities, AIAA Journal, Vol.26, pp.195–203, 1988.
4) Hertzberg, A., Bruckner, A.P., and Knowlen, C., Experimental Investigation of Ram Accelerator Propulsion Modes, Shock Waves, Vol.1, pp.17–25, 1991.
5) Kaloupis, P. and Bruckner, A.P., The Ram Accelerator: A Chemically Driven Mass Launcher, AIAA-88-2968, 1988.

6) Li, C., Kailasanath, K., Oran, E.S., Landsberg, A.M., and Boris, J.P., Stability of Oblique Detonations in Ram Accelerators, AIAA-92-0089, 1992.
7) Yungster, S. and Bruckner, A.P., Computational Studies of a Superdetonative Ram Accelerator Mode, J. Propulsion and Power, Vol.8, pp.457–463, 1992.
8) Oran, E.S. and Boris, J.P., Numerical Simulation of Reactive Flow, Elsevier, NY, 1987.
9) Shinn, J.L., Yee, H.C., and Uenishi, K., Extension of a Semi-Implicit Shock-Capturing Algorithm for 3-D Fully Coupled, Chemically Reacting Flows in Generalized Coordinates, AIAA-87-1577, 1987.
10) Liu, Y. and Vinokur, M., Upwind Algorithms for General Thermo-Chemical Nonequilibrium Flows, AIAA-89-0201, 1989.
11) Harten, A., High Resolution Schemes for Hyperbolic Conservation Laws, J. Comp. Physics, Vol.49, pp.357–393, 1983.
12) Yee, H.C., Construction of Explicit and Implicit Symmetric TVD Schemes and their Applications, J. Comp. Physics, Vol.68, pp.151–179, 1987.
13) Matsuo, A., Fujiwara, T., and Fujii, K.; Key Parameters of the Periodic Unsteadiness Around the Projectile Travelling at Hypervelocities, Proceedings of the 14th ICDERS, Coimbra, Portugal, 1994.
14) Matsuo, A., Characteristics of the Standing Oblique Detonation Supported by a Two-Dimensional Blunted Wedge and an Axisymmetric Blunt Body, phD dissertation, University of Nagoya, Japan, 1993.
15) Matsuo, A. and Fujiwara, T., Numerical Investigation of Oscillatory Instability in Shock-Induced Combustion around a Blunt Body, AIAA Journal, Vol.31, pp.1835–1841, 1993.
16) Li, C., Kailasanath, K., and Oran, E.S., Structure of Reaction Waves behind Oblique Shocks, Prog. Astro. and Aero., Vol.153, pp.231–241, 1993.
17) Weber, J.W., The Efficient Simulation of Gas Detonation Physics on a Massively Parallel Connection Machine, phD Dissertation, University of Maryland, UM-Aero 94-01, 1994.
18) Lefebvre, M.H., Oran, E.S., Kailasanath, K., and Van Tiggelen, P.J., The Influence of the Heat Capacity and Diluent on Detonation Structure, Comb. and Flame, Vol.95, pp.206–218, 1993.
19) Korobeinikov, V.P., Levin, V.A., Markov, V.V., and Chernyi, G.G., Propagation of Blast Waves in a Combustible Gas, Astronautica Acta, Vol.17, pp.529–537, 1972.
20) Lefebvre, M.H., Oran, E.S., Kailasanath, K., and Van Tiggelen, P.J., Twodimensional Modeling of Gaseous Detonation, Prog. Astro. and Aero., Vol 153, pp.64–77, 1993.
21) Wilson, G.J. and MacCormack, R.W., Modeling Supersonic Combustion Using a Fully-Implicit Numerical Method, AIAA-90-2307, 1990.
22) Jachimowski, C.J., An Analytical Study of the Hydrogen-Air Reaction Mechanism with Application to Scramjet Combustion, NASA TP 2791, 1988.
23) Drummond, J.P., Numerical Study of a Scramjet Engine Flow Field, AIAA-81-185, 1981.
24) Hirsch, C., Numerical Computation of Internal and External Flows, Vol.1: Fundamentals of Numerical Discretization, John Wiley & Sons, 1991.
25) Hirsch, C., Numerical Computation of Internal and External Flows, Vol.2: Computational Methods for Inviscid and Viscous Flows, John Wiley & Sons, 1991.
26) Boris, J.P. and Book, D.L., Solution of the Continuity Equation by the Method of Flux Corrected Transport, J. Comp. Physics, Vol.16, pp.85-129, 1976.
27) Salezak, S.T., Fully Multidimensional Flux-Corrected-Transport, J. Comp. Physics, Vol.31, pp.355–362, 1979.
28) Van Leer, B., Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method, J. Comp. Physics, Vol.32, pp.101–136, 1979.
29) Yee, H.C., Upwind and Symmetric Shock Capturing Schemes, NASA TM–89464, 1987.
30) Harten, A. and Hyman, J.M., A Self-Adjusting Grid for the Computation of Weak Solutions of Hyperbolic Conservation Laws, J. Comp. Physics, Vol.50, pp.235–269, 1983.
31) Roe, P.L., Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, J. Comp. Physics, Vol.43, pp.357–372, 1981.
32) Yee, H.C. and Shinn, J.L., Semi-Implicit and Fully-Implicit Shock Capturing Methods for Hyperbolic Conservation Laws with Stiff Source Terms, AIAA-87-1116, 1987.

33) Yee, H.C. and Harten, A., Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates, AIAA Journal, Vol.25, pp.266–274, 1987.
34) Fujiwara, T., Matsuo, A., and Nomoto, H., A Two-Dimensional Detonation Supported by a Blunt Body or a Wedge, AIAA-88-0098, 26th Aerospace Sciences Meeting, Reno, NV, 1988.
35) Li, C., Kailasanath, K., Oran, E.S., Landsberg, A.M., and Boris, J.P., Analysis of Transient Flows in Thermally Choked Ram Accelerators, AIAA-93-2187, 1993.
36) Nusca, M.J., Numerical Simulation of Fluid Dynamics with Finite-Rate and Equilibrium Combustion Kinetics for the 120 mm Ram Accelerator, AIAA-93-2182, 1993.
37) Soetrismo, M., Imlay, S.T., and Roberts, D.W., Numerical Simulations of the Superdetonative Ram Accelerator Combusting Flow Field, AIAA-93-2185, 1993.
38) Bruckner, A.P. and Knowlen, C., Overview of Ram Accelerator Technology, National Shock Wave Symposium, Sendai, Japan, 1993.