Memoirs of the Faculty of Engineering, Nagoya University
Vol. 41, No. 1 (1989)

"HYREFS SERIES CODES" USERS' MANUAL

K. VISWANATH REDDY and TOSHI FUJIWARA

Department of Aeronautical Engineering,

(Received May 31, 1989)

Abstract

HYpersonic REacting Flow Solutions can be generated using these HYREFS computer codes for a wide range of Mach numbers and angles of attack. These programs are designed to visualize various aspects of the nonequilibrium reacting axisymmetric as well as three-dimensional flows around a blunt-nosed configuration. The versatility of these codes in handling perfect gas, calorically imperfect but chemically frozen gas, and chemically reacting gas can be utilized to demonstrate the importance and limitations of each gas model. Basically these programs use a Sphere-Cone-Cylinder body; hence, one can easily change the geometry by altering the lengths of each section. This manual briefly describes the methodology used for simulating these complex flows and explains in detail how to use HYREFS codes. Some programming aspects are also indicated. The code structure and the various operations carried out by the subroutines are detailed and the input-output features are explained. Some sample calculations for viscous reacting flows are presented.

HYREFS were developed primarily as a research tool to establish accurate solutions for the flows around reentry vehicles. The governing equations are solved by using a powerful implicit scheme and the implementation of boundary conditions is made in a compatible way. As a result, in several test problems high degree of convergence is achieved. Many vectorization techniques are used to touch the maximum machine's MFLOPS (Mega FLOating Point operations per Second) rate.

Contents

1. Introduction

Recent space operations have triggered renewed interest for the aerothermodynamicists to conduct research in nonequlibrium flows. When a vehicle is traveling at hypersonic speeds at a high altitude where air density is low, flow behind the normal shock that is formed over the vehicle will tend to be out of equilibrium both chemically and thermally. In the nonequilibrium regime of the flow, thermal energy of the gas will be higher than its equilibrium value because the absorption of energy due to dissociation does not take place to the extent dictated by the equilibrium conditions. As suggested by Park[1] a thermally and chemically nonequilibrium flow can be analysed by introducing several temperatures[2], one for each energy mode. This not only increases the number of equations of the already swollen system of governing equations with the species conservation equations, but also various transport properties appear in the new energy equations.[3] As a result, works on thermally nonequilibrium flows are still premature. On the other hand, a thermally equilibrium but chemically nonequilibrium gas model can predict the important quantities such as heat transfer, shock locations, skin-friction etc.,[4,5] quite accurately in most of the practical hypersonic flows.

HYREFS codes represent a series of codes that can be used for HYpersonic REacting Flow Simulations. They include five packages:

- HYREFS-NR2. A two-dimensional nonreacting code. Can handle perfect gas and frozen chemistry gas models.
- HYREFS-CR2. A two-dimensional chemically reacting code. Can handle frozen chemistry gas and reacting gas models.
- HYREFS-GR2. A gaphics code for two-dimensional flows.
- HYREFS-NR3. A three-dimensional nonreacting code. Can handle perfect gas and frozen chemistry gas models.
- HYREFS-CR3. A three-dimensional chemically reacting code.
- HYREFS-GR3. A gaphics code for three-dimensional flows.

Besides the capability of handling different gas models, all the above codes can provide viscous and inviscid solutions. The striking feature of these codes is that they are made user-friendly and any option can be activated just by choosing appropriate values for the input variables. These are explained in detail in the Input-Output section. HYREFS codes use thermally perfect but calorically imperfect multicomponent gas mixtures as flow media. Five species (Oxygen molecule, Nitrogen molecule, Oxygen atom, Nitrogen atom and Nitric oxide molecule) are considered and the five most important chemical reactions among these constituent species are allowed. The resulting five species conservation equations are solved in a fully coupled manner along with the three Navier-Stokes equations and an energy equation using a noniterative, approximately-factorized implicit scheme. An axisymmetric flow can also be generated using a three-dimensional full-fledged code by setting angle of attack equal to zero. But, since some users will be interested only in these two-dimensional flows, to make such calculations efficient, equivalent two-dimensional codes are also developed. All the discussions in the following sections are for the three-dimensional flows. However, usage about two-dimensional code is pointed out wherever it is different from three-dimensional flows.

2. Solution Algorithm

The methodology employed for solving these reacting flows is briefly explained here. Since these codes are developed over some period of time, there could be some minor

deviations in the codes from the mathematics presented in the following sections. And also a few complicated mathematical expressions have been avoided to make this text relatively simple. However, additional details can be obtained from the listed references and the codes themselves.

2. 1. *Governing Equations*

A complete system of governing equations that describe a three-dimensional, viscous flow of a multicomponent reacting gas is obtained[6] after assuming that (1) the flow is in thermal equilibrium, (2) the radiation and body forces are of negligible order, and (3) the molecular diffusion is due to binary only. The basic cylindrical coordinate system (t, z, r, θ) is used. These equations are then transformed into a general moving body-oriented curvilinear coordinate system (τ,ξ,η,ζ) and then rearranged into a strong conservation-law form as follows:

$$\frac{\partial \hat{q}}{\partial t}+\frac{\partial \hat{E}}{\partial \xi}+\frac{\partial \hat{F}}{\partial \eta}+\frac{\partial \hat{G}}{\partial \zeta}=\frac{1}{Re}\left[\frac{\partial \hat{E}_v}{\partial \xi}+\frac{\partial \hat{F}_v}{\partial \eta}+\frac{\partial \hat{G}_v}{\partial \zeta}\right]+\hat{H}+\hat{H}_1 . \tag{1}$$

where the (N_s+4) component inviscid and viscous fluxes are

$$\hat{q}=J^{-1}\begin{vmatrix}\rho_i\\ \rho u\\ \rho u\\ \rho w\\ e\end{vmatrix};\quad \hat{E}=J^{-1}\begin{vmatrix}\rho_i U\\ \rho uU+\xi_z p\\ \rho vU+\xi_r p\\ \rho wU+\xi_\theta p/r\\ (e+p)U-\xi_t p\end{vmatrix};$$

$$\hat{F}=J^{-1}\begin{vmatrix}\rho_i V\\ \rho uV+\eta_z p\\ \rho vV+\eta_r p\\ \rho wV+\eta_\theta p/r\\ (e+p)V-\eta_t p\end{vmatrix};\quad \hat{G}=J^{-1}\begin{vmatrix}\rho_i W\\ \rho uW+\zeta_z p\\ \rho vW+\zeta_r p\\ \rho wW+\zeta_\theta p/r\\ (e+p)W-\zeta_t p\end{vmatrix};$$

$$\hat{E}_v=J^{-1}\begin{vmatrix}-\rho_i\beta_3(\xi_z\tilde{u}_i+\xi_r\tilde{u}_i+\xi_\theta\tilde{w}_i/r)\\ \xi_z\tau_{zz}+\xi_r\tau_{zr}+\xi_\theta\tau_{z\theta}/r\\ \xi_z\tau_{zr}+\xi_r\tau_{rr}+\xi_\theta\tau_{r\theta}/r\\ \xi_z\tau_{z\theta}+\xi_r\tau_{r\theta}+\xi_\theta\tau_{\theta\theta}/r\\ \xi_z\beta_z+\xi_r\beta_r+\xi_\theta\beta_\theta/r\end{vmatrix};$$

$$\hat{F}_v = J^{-1} \begin{vmatrix} -\rho_i \beta_3 (\eta_z \tilde{u}_i + \eta_{r\tilde{v}i} + \eta_\theta \tilde{w}_i / r) \\ \eta_z \tau_{zz} + \eta_r \tau_{zr} + \eta_\theta \tau_{z\theta} / r \\ \eta_z \tau_{zr} + \eta_r \tau_{rr} + \eta_\theta \tau_{r\theta} / r \\ \eta_z \tau_{z\theta} + \eta_r \tau_{r\theta} + \eta_\theta \tau_{\theta\theta} / r \\ \eta_z \beta_z + \eta_r \beta_r + \eta_\theta \beta_\theta / r \end{vmatrix};$$

$$\hat{G}_v = J^{-1} \begin{vmatrix} -\rho_i \beta_3 (\zeta_z \tilde{u}_i + \zeta_r \tilde{v}_i + \zeta_\theta \tilde{w}_i / r) \\ \zeta_z \tau_{zz} + \zeta_r \tau_{zr} + \zeta_\theta \tau_{z\theta} / r \\ \zeta_z \tau_{zr} + \zeta_r \tau_{rr} + \zeta_\theta \tau_{r\theta} / r \\ \zeta_z \tau_{z\theta} + \zeta_r \tau_{r\theta} + \zeta_\theta \tau_{\theta\theta} / r \\ \zeta_z \beta_z + \zeta_r \beta_r + \zeta_\theta \beta_\theta / r \end{vmatrix};$$

$$\hat{H} = J^{-1} \begin{vmatrix} \dot{W}_i \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix};$$

$$\hat{H}_1 = \frac{J^{-1}}{r} \begin{vmatrix} -\rho_i v - \rho_i \tilde{v}_i \beta_3 / Re \\ -\rho u v + \tau_{zr} / Re \\ -\rho (v^2 - w^2) + (t_{rr} - \tau_{\theta\theta}) / Re \\ -2 \rho v w + 2 t_{ry} / Re \\ -(e + p) v + (\tau_{zr} u + \tau_{rr} v + \tau_{r\theta} w - q_r) / Re \end{vmatrix};$$

with the contravariant velocities defined as

$$U = \xi_t + \xi_z u + \xi_r v + \xi_\theta w / r ,$$

$$V = \eta_t + \eta_z u + \eta_r v + \eta_\theta w / r ,$$

$$W = \zeta_t + \zeta_z u + \zeta_r v + \zeta_\theta w / r .$$

The viscous and thermal stresses are given by

$$\tau_{zz} = \mu [\frac{4}{3} \frac{\partial u}{\partial z} - \frac{2}{3} \frac{\partial v}{\partial r} - \frac{2}{3} \frac{1}{r} \frac{\partial w}{\partial \theta} - \frac{2}{3} \frac{v}{r}],$$

$$\tau_{rr} = \mu\left[-\frac{2}{3}\frac{\partial u}{\partial z} + \frac{4}{3}\frac{\partial v}{\partial r} - \frac{2}{3}\frac{1}{r}\frac{\partial w}{\partial \theta} - \frac{2}{3}\frac{v}{r}\right],$$

$$\tau_{\theta\theta} = \mu\left[-\frac{2}{3}\frac{\partial u}{\partial z} - \frac{2}{3}\frac{\partial v}{\partial r} + \frac{4}{3}\left(\frac{1}{r}\frac{\partial w}{\partial \theta} + \frac{v}{r}\right)\right],$$

$$\tau_{zr} = \tau_{rz} = \mu\left[\frac{\partial u}{\partial r} + \frac{\partial v}{\partial z}\right],$$

$$\tau_{r\theta} = \tau_{\theta r} = \mu\left[\frac{\partial w}{qr} + \frac{1}{r}\frac{\partial v}{\partial \theta} - \frac{w}{r}\right],$$

$$\tau_{z\theta} = \tau_{\theta z} = \mu\left[\frac{\partial w}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial \theta}\right],$$

$$\beta_z = u\tau_{zz} + v\tau_{zr} + w\tau_{z\theta} - q_z,$$

$$\beta_r = u\tau_{zr} + v\tau_{rr} + w\tau_{\theta r} - q_r,$$

$$\beta_\theta = u\tau_{z\theta} + u\tau_{r\theta} + w\tau_{\theta\theta} - q_\theta,$$

$$q_z = -\beta_2\kappa\frac{\partial T}{\partial z} + \beta_3\sum_i^{N_s}\rho_i h_i \tilde{u}_i,$$

$$q_r = -\beta_2\kappa\frac{\partial T}{\partial r} + \beta_3\sum_i^{N_s}\rho_i h_i \tilde{v}_i,$$

$$q_\theta = -\beta_2\kappa\frac{1}{r}\frac{\partial T}{\partial \theta} + \beta_3\sum_i^{N_s}\rho_i h_i \tilde{w}_i.$$

The following nondimensionalization has been employed in the above formulations:

$$z, r = \frac{z^*, r^*}{L^*}; \quad u, v, w = \frac{u^*, v^*, w^*}{V_\infty^*}; \quad t = \frac{t^*}{L^*}V_\infty^*;$$

$$\rho = \frac{\rho^*}{\rho_\infty^*}; \quad p = \frac{p^*}{\rho_\infty^* V_\infty^{*2}}; \quad e = \frac{e^*}{\rho_\infty^* V_\infty^{*2}};$$

$$T = \frac{T^*}{T_\infty^*}; \quad h = \frac{h^*}{V_\infty^{*2}}; \quad \tilde{u}, \tilde{v}, \tilde{w} = \frac{\tilde{u}^*, \tilde{v}^*, \tilde{w}^*}{D_\infty^*}L^*;$$

$$\dot{W} = \frac{\dot{W}^*}{\rho_\infty^* V_\infty^*}L^*; \quad \mu = \frac{\mu^*}{\mu_\infty^*}; \quad \kappa = \frac{\kappa^*}{\kappa_\infty^*};$$

$$D = \frac{D^*}{D_\infty^*}; \quad M = \frac{M^*}{M_\infty^*}; \quad c_p = \frac{c_p^*}{V_\infty^{*2}}T_\infty^*.$$

Here the superscript * represents dimensional variables while the subscript ∞ indicates free-stream values. The constants appearing due to nondimensionalization are

$$Re = \frac{\rho_\infty^* V_\infty^* L^*}{\mu_\infty^*} \; ; \quad \beta_1 = \frac{\Re_0 T_\infty^*}{M_\infty^* V_\infty^{*2}} \; ;$$

$$\beta_2 = \frac{T_\infty^* \kappa_\infty^*}{\mu_\infty^* V_\infty^{*2}} \; ; \quad \beta_3 = \frac{\rho_\infty^* D_\infty^*}{\mu_\infty^*} \; .$$

Here it is understood that the cylindrical-coordinate derivatives are to be expanded in (ξ, η, ζ) system via chain rule such as

$$u_z = \xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta \; , \quad v_z = \xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta \; ,$$

$$w_z = \xi_z w_j + n_z w_n + z_z w_z \; .$$

The global species continuity can be written as

$$\rho = \sum_{i=1}^{N_s} \rho_i \; . \tag{2}$$

The following state equations are also made use of:

$$p = \sum_{i=1}^{N_s} \rho_i h_i + \frac{1}{2} \rho (u^2 + v^2 + \omega^2) - e \; , \tag{3}$$

$$p = \rho \beta_1 T \sum_{i=1}^{N_s} (f_i / M_i) \; , \tag{4}$$

where f_i is the mass fraction of i-th species.

Diffusion velocities $\tilde{u}, \tilde{v}, \tilde{w}$ can be obtained accurately for a multicomponent mixtures of N_s gases by solving the multicomponent equations. Since solving such equations requires a considerable cpu time, diffusion velocities in these codes are calculated using Fick's law which represents multicomponent equations in a binary-mixture of gases as follows:

$$\tilde{u} = -\frac{D_i}{f_i} \frac{\partial f_i}{\partial z} \; ; \quad \tilde{v} = -\frac{D_i}{f_i} \frac{\partial f_i}{\partial r} \; ; \quad \tilde{w} = -\frac{1}{r} \frac{D_i}{f_i} \frac{\partial f_i}{\partial \theta} \; . \tag{5}$$

2. 1. 1. Chemical kinetics

In the present work, gas model is assumed to be a mixture of five calorically imperfect species, namely molecular oxygen O_2, molecular nitrogen N_2, atomic nitrogen N, atomic oxygen O and nitric oxide NO. Only the five most important chemical reactions that may take place among these five constituent species are considered. Symbolically they are represented with the following equations:

$$N_2 + M \underset{k_{b_1}}{\overset{k_{f_1}}{\rightleftharpoons}} N + N + M ,$$

$$O_2 + M \underset{k_{b_2}}{\overset{k_{f_2}}{\rightleftharpoons}} O + O + M ,$$

$$NO + M \underset{k_{b_3}}{\overset{k_{f_3}}{\rightleftharpoons}} N + O + M ,$$

$$O + N_2 \underset{k_{b_4}}{\overset{k_{f_4}}{\rightleftharpoons}} NO + N ,$$

$$NO + O \underset{k_{b_5}}{\overset{k_{f_5}}{\rightleftharpoons}} N + O_2 ,$$

where M can be any other molecule available in the gas mixture. The forward reaction rate of j-th reaction is given by the modified Arrhenius law

$$k_{f_j} = S_j T^{*\beta_j} exp\{\frac{-E_{aj}}{\Re_0 T^*}\} . \tag{6}$$

The values of S_j and E_{a_j} are given in Table 1. The backward reaction rate is obtained from the relation

$$k_{b_j} = \frac{k_{f_j}}{K_{C_j}} , \tag{7}$$

where K_{C_j} is the concentration equilibrium constant. An empirical formula

$$K_{C_j} = exp(A_{1_j} + A_{2_j} \ln Z + A_{3_j} Z + A_{4_j} Z^2 + A_{5_j} Z^3) , \tag{8}$$

$$Z = \frac{10000}{T^*} , \tag{9}$$

is used to determine the values for K_{C_j} at any temperature between 293 K and 50,000 K. All the constants (A_{1_j} to A_{5_j}) appearing in this equation are listed in Table 1. Once the forward and backward reaction rates have been determined, the mass production rate of each species is calculated using the law of mass action. For the general chemical reaction

$$\sum_{i=1}^{N_s} \vartheta'_{ji} C_i \underset{k_{b_j}}{\overset{k_{f_j}}{\rightleftharpoons}} \sum_{i=1}^{N_s} \vartheta''_{ji} C_i \qquad j = 1, N_r , \tag{10}$$

Table 1. Constants for Modified Arrhenius Approximations of Forward Reaction-Rates and for Exponential Approximations of Equilibrium Constants.

Reaction	S_i $\frac{cm^3}{mole.sec}$	β_j	E_{a_j}	A_{1_i}	A_{2_i}	A_{3_i}	A_{4_i}	A_{5_i}
O_2 +M ↔ O+O+M	M=N_2 2.75E19 M=N 8.25E19 M=O_2 2.75E19 M=O 8.25E19 M=NO 2.75E19	-1.0	59500	2.855	0.988	-6.181	-0.023	-0.001
N_2 +M ↔ N+N+M	M=N_2 3.70E21 M=N 1.11E21 M=O_2 3.70E21 M=O 1.11E21 M=NO 3.70E21	-1.6	113200	1.858	-1.325	-9.856	-0.174	0.008
NO +M ↔ N+O+M	M=N_2 2.30E17 M=N 4.60E17 M=O_2 2.30E17 M=O 4.60E17 M=NO 2.30E17	-0.5	75500	0.792	-0.492	-6.761	-0.091	0.004
O +N_2 ↔ N+NO	3.18E13	0.1	37700	1.066	-0.833	-3.095	-0.084	0.004
NO +O ↔ N+O_2	2.16E08	1.29	19220	-2.063	-1.480	-0.580	-0.114	0.005

the law of mass action gives the rate of production of i-th species due to j-th reaction as follows:

$$(\dot{W}_i^*)_j = (\vartheta''_{ji} - \vartheta'_{ji})\, M_i^* \left[k_{f_j} \prod_{l=1}^{N_s} C_l^{\vartheta'_{jl}} - k_{b_j} \prod_{l=1}^{N_s} C_l^{\vartheta''_{jl}} \right]. \qquad (11)$$

The net rate of mass production of i-th species due to all the N_r reactions is then found by summing the contributions from each reaction

$$\dot{W}_i^* = \sum_{j=1}^{N_r} (\dot{W}_i^*)_j . \qquad (12)$$

2. 1. 2. Thermodynamic model

The enthalpy and specific heat of the mixture are obtained from the following expressions[7]:

$$h^* = \sum_{i=1}^{N_s} f_i h_i^* , \tag{13}$$

$$c_p^* = \sum_{i=1}^{N_s} f_i c_{pi}^* , \tag{14}$$

where h_i^* and c_{pi}^* are the dimensional enthalpy and specific heat of the i-th species. The values for h_i^* and c_{pi}^* as functions of temperature for the constituent species are obtained from the polynomial curve fittings developed by Moss et al. They are as follows:
Specific heat:

$$\frac{c_{pi}^*}{\mathfrak{R}_0} = a_{1_i} + a_{2_i} T^* + a_{3_i} T^{*2} + a_{4_i} T^{*3} + a_{5_i} T^{*4} . \tag{15}$$

Enthalpy:

$$\frac{h_i^*}{\mathfrak{R}_0 T^*} = a_{1_i} + \frac{a_{2_i} T^*}{2} + \frac{a_{3_i} T^{*2}}{3} + \frac{a_{4_i} T^{*3}}{4} + \frac{a_{5_i} T^{*4}}{5} + \frac{a_{6_i}}{T^*} . \tag{16}$$

The constants a_{1_i} to a_{6_i} for the five species considered in this study are tabulated in Tables 2 through 4.

Table 2. Constants in Polynomial Curve Fitting of Thermodynamic Properties.

Species	*Coefficients*					
	a_{1_i}	a_{2_i}	a_{3_i}	a_{4_i}	a_{5_i}	a_{6_i}
0 to 1000 K						
O_2	3.62560	-1.87822 x 10^{-3}	7.05545 x 10^{-6}	-6.76351 x 10^{-9}	2.15560 x 10^{-12}	-1.04752 x 10^{3}
N_2	3.67483	-1.20815 x 10^{-3}	2.32401 x 10^{-6}	-6.32176 x 10^{-10}	-2.25773 x 10^{-13}	-1.06116 x 10^{3}
N	2.50307	-2.18002 x 10^{-5}	5.42053 x 10^{-8}	-5.64756 x 10^{-11}	2.09990 x 10^{-14}	5.60989 x 10^{4}
O	2.94643	-1.63817 x 10^{-3}	2.42103 x 10^{-6}	-1.60284 x 10^{-9}	3.89070 x 10^{-13}	2.91476 x 10^{4}
NO	4.04595	-3.41818 x 10^{-3}	7.98192 x 10^{-6}	-6.11393 x 10^{-9}	1.59191 x 10^{-12}	9.74539 x 10^{3}
1000 to 6000 K						
O_2	3.62195	7.36183 x 10^{-4}	-1.96522 x 10^{-7}	3.62016 x 10^{-11}	-2.89456 x 10^{-15}	-1.20198 x 10^{3}
N_2	2.89632	1.51549 x 10^{-3}	-5.72353 x 10^{-7}	9.98074 x 10^{-11}	-6.52236 x 10^{-15}	-9.05862 x 10^{2}
N	2.45027	1.06615 x 10^{-4}	-7.46534 x 10^{-8}	1.87965 x 10^{-11}	-1.02598 x 10^{-15}	5.61160 x 10^{4}
O	2.54206	-2.75506 x 10^{-5}	-3.10280 x 10^{-9}	4.55107 x 10^{-12}	-4.36805 x 10^{-16}	2.92308 x 10^{4}
NO	3.18900	1.33823 x 10^{-3}	-5.28993 x 10^{-7}	9.59193 x 10^{-11}	-6.48479 x 10^{-15}	9.82833 x 10^{3}

Table 3. Constants in Polynomial Curve Fitting of Thermodynamic Properties

Species	*Coefficients*					
	a_{1_i}	a_{2_i}	a_{3_i}	a_{4_i}	a_{5_i}	a_{6_i}
6000 to 15000 K						
O_2	3.72100	4.25400 x 10^{-4}	-2.83500 x 10^{-8}	6.05000 x 10^{-13}	-5.18600 x 10^{-18}	-1.04400 x 10^{3}
N_2	3.72700	4.68400 x 10^{-4}	-1.14000 x 10^{-7}	1.15400 x 10^{-11}	-3.29300 x 10^{-16}	-1.04300 x 10^{3}
N	2.74800	-3.90900 x 10^{-4}	1.33800 x 10^{-7}	-1.19100 x 10^{-11}	3.36900 x 10^{-16}	5.60900 x 10^{4}
O	2.54600	-5.95200 x 10^{-5}	2.70100 x 10^{-8}	-2.79800 x 10^{-12}	9.38000 x 10^{-17}	2.91500 x 10^{4}
NO	3.84500	2.52100 x 10^{-4}	-2.65800 x 10^{-8}	2.16200 x 10^{-12}	-6.38100 x 10^{-17}	9.76400 x 10^{3}
15000 to 25000 K						
O_2	3.48666	5.23842 x 10^{-4}	-3.91234 x 10^{-8}	1.00935 x 10^{-12}	-8.87183 x 10^{-18}	-1.04400 x 10^{3}
N_2	9.63769	-2.57284 x 10^{-3}	3.30198 x 10^{-7}	-1.43149 x 10^{-11}	2.03326 x 10^{-16}	-1.04300 x 10^{3}
N	-1.22799	1.92685 x 10^{-3}	-2.43705 x 10^{-7}	1.21930 x 10^{-11}	-1.99184 x 10^{-16}	5.60900 x 10^{4}
O	-0.00979	1.24497 x 10^{-3}	-1.61544 x 10^{-7}	8.03799 x 10^{-12}	-1.26240 x 10^{-16}	2.91500 x 10^{4}
NO	4.33087	-5.80863 x 10^{-5}	2.80595 x 10^{-8}	-1.56941 x 10^{-12}	2.41039 x 10^{-17}	9.76400 x 10^{3}

Table 4. Constants in Polynomial Curve Fitting of Thermodynamic Properties (Cond.).

Species	*Coefficients*					
	a_{1_i}	a_{2_i}	a_{3_i}	a_{4_i}	a_{5_i}	a_{6_i}
25000 to 35000 K						
O_2	3.96198	3.94455 x 10^{-4}	-2.95058 x 10^{-8}	7.39745 x 10^{-13}	-6.42093 x 10^{-18}	-1.04400 x 10^{3}
N_2	-5.16808	2.33369 x 10^{-3}	-1.29534 x 10^{-7}	2.78721 x 10^{-12}	-2.13596 x 10^{-17}	-1.04300 x 10^{3}
N	15.5202	-3.88579 x 10^{-3}	3.22884 x 10^{-7}	-9.60527 x 10^{-12}	9.54722 x 10^{-17}	5.60900 x 10^{4}
O	16.4281	-3.93130 x 10^{-3}	2.98399 x 10^{-7}	-8.16128 x 10^{-12}	7.50043 x 10^{-17}	2.91500 x 10^{4}
NO	2.35075	5.86430 x 10^{-4}	-3.13165 x 10^{-8}	6.04951 x 10^{-13}	-4.05567 x 10^{-18}	9.76400 x 10^{3}

2. 1. 3. Transport models

In the governing equations three types of transport coefficients μ, κ and D_i are appearing. They represent viscous, thermal and molecular diffusion flux transports respectively. Values for these coefficients have to be estimated at the corresponding temperatures. The following methodology has been used in these codes :

Individual species viscosity is obtained from Sutherland's law

$$\frac{\mu_i^*}{\mu_{i0}^*} = \left(\frac{T^*}{T_{i0}^*}\right)^{3/2} \left(\frac{T_{i0}^* + S_i}{T^* + S_i}\right), \qquad (17)$$

Table 5. Constants Used in Various Transport Equations.

Species	*Viscosity*			*Thermal Conductivity*			*Molecular*		
	T_0 Rankine	S Rankine	μ_0 milliPoise	T_0' Rankine	S' Rankine	κ_0 $\frac{Btu}{(h)(ft)(R)}$	T_ε K	σ A	M $\frac{g}{mole}$
O_2	491.69	250.00	0.1919	491.69	400.0	0.01419	106.7	3.467	32.00
N_2	491.69	192.00	0.1663	491.69	300.0	0.014	71.4	3.798	28.02
N	491.69	250.00	0.1919	491.69	400.0	0.01419	71.4	3.298	14.01
O	491.69	250.00	0.1919	491.69	400.0	0.01419	106.7	3.050	16.00
NO	491.69	250.00	0.1919	491.69	400.0	0.01419	116.7	3.492	30.01

where μ_{i0}^* and T_{i0}^* are the reference values and S_i the Sutherland constant for i-th species. All these consants are taken from Reference 8 and are also tabulated in Table 5. Once the viscosity of each species is determined, the mixture viscosity is estimated by using Wilke's law

$$\mu^* = \sum_{i=1}^{N_s} \frac{\mu_i^*}{1 + \frac{1}{X_i} \sum_{\substack{j=1 \\ j \neq i}}^{N_s} X_j \phi_{ij}}, \tag{18}$$

where

$$\phi_{ij} = \frac{\{1 + [(\mu_i^*/\mu_j^*)(\rho_j/\rho_i)]^{1/2} (M_i/M_j)^{1/4}\}^2}{\frac{4}{\sqrt{2}}[1 + (M_i/M_j)]^{1/2}}. \tag{19}$$

The species thermal conductivity is also obtained from the Sutherland second law

$$\frac{\kappa_i^*}{\kappa_{i0}^*} = \left(\frac{T^*}{T_{i0}^{*\prime}}\right)^{3/2} \left(\frac{T_{i0}^{*\prime} + S_i'}{T^* + S_i'}\right). \tag{20}$$

The values of κ_{i0}^*, $T_{i0}^{*\prime}$ and the second Sutherland constant S_i' are also obtained from Reference 8 and are given in Table 5. The mixture thermal conductivity is obtained using Wassilewas' formula

$$\kappa^* = \sum_{i=1}^{N_s} \frac{\kappa_i^*}{1 + \frac{1}{X_i} \sum_{\substack{j=1 \\ j \neq i}}^{N_s} X_j \phi'_{ij}}, \tag{21}$$

where ϕ'_{ij}=1.065 ϕ_{ij}. The binary self-diffusion coefficient D_i^* for the i-th species is determined from the emprical formula given by Chapman and Cowling

$$D_i^* = \frac{0.001858 T^{*3/2} \sqrt{2 M_i^*}}{p^* \sigma_i^2 \Omega_D}. \tag{22}$$

Here the diffusion collision integral is approximated by

$$\Omega_D = T_*^{-0.145} + (T_* + 0.5)^{-2}, \tag{23}$$

with $T_* = T^*/T_{\epsilon i}$. Values of the effective temperature $T_{\epsilon i}$ and effective collision diameter σ are taken from Reference 8. These values are given in Table 5.

2. 2. *Grid System*

The simple geometry for a rocket is shown in Fig. 1. This configuration consists of a spherical head and a long cylindrical body. The head and body are smoothly connected with a conoid. By varying the sizes of each portion, several configurations can be generated. For example, by letting zero length for the conical portion a blunted cylinder configuration can be generated. A floating body-oriented grid system is constructed using ξ, η and ζ space coordinates. It consists of fixed radial rays (ξ=constant and ζ=constant) and floating η=constant surfaces. The innermost η=0 surface coincides with the body surface and the outermost one always coincides with the moving bow shock surface. All the other η=constant grid surfaces filling the shock layer are distributed exponentially between these two boundary surfaces. Exponential stretching is used to allow grid points to concentrate near the body surface and the peripheral shock surface. The equations used for transforming η^* coordinate to η are given by

$$\eta = g_1(e^{\gamma_1 \eta^*} - 1), \quad \text{for } 0 \le \eta^* \le \eta_c^*, \tag{24}$$

$$\eta = 1 + g_2[1 - e^{\gamma_2(1-\eta^*)}], \quad \text{for } \eta_c^* \le \eta^* \le 1, \tag{25}$$

where g_1, g_2, γ_1 and γ_2 are the constants to be determined from the given grid spacings near the body and near the shock wave and η^* represents the point at which both of the algebraic expressions should give the same value. In the present simulations half of the grid points are made to lie between η=0 and $\eta=\eta_c^*$, and the other half lie between $\eta=\eta_c^*$ and η=1. The user has to specify $\Delta\eta_{body}$, $\Delta\eta_{shock}$ and η_c^* as input variables. Grid points in the ξ coordinate direction are smoothly distributed on the entire body as shown in Fig. 2 while in the meridian plane they are uniformly distributed.

2. 3. *Numerical Scheme*

The governing equations are approximated by adopting the non-iterative approximately-factorized implicit method of Beam and Warming[9]. The scheme can be wirtten in the following three steps:

$$\begin{aligned}(I + \Delta\tau\delta_\xi\hat{A}^n - \Delta\tau Re^{-1}\delta_\xi J^{-1}\hat{P}^n J - \varepsilon_I J^{-1}\nabla_\xi\Delta_\xi J)\Delta\tilde{q} \\ = -\Delta\tau[\delta_\xi\hat{E}^n + \delta_\eta\hat{F}^n + \delta_\zeta\hat{G}^n \\ - Re^{-1}(\delta_\xi\hat{E}_v^n + \delta_\eta\hat{F}_v^n + \delta_\zeta\hat{G}_v^n) - \hat{H}^n] \\ - \varepsilon_E J^{-1}[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\zeta\Delta_\zeta)^2] J\hat{q},\end{aligned} \tag{26}$$

$$(I + \Delta\tau\delta_\eta\hat{B}^n - \Delta\tau Re^{-1}\delta_\eta J^{-1}\hat{Q}^n J - \varepsilon_I J^{-1}\nabla_\eta\Delta_\eta J)\Delta\tilde{q} = \Delta\tilde{q}, \tag{27}$$

$$(I + \Delta\tau\delta_\zeta\hat{C}^n - \Delta\tau Re^{-1}\delta_\zeta J^{-1}\hat{R}^n J - \varepsilon_I J^{-1}\nabla_\zeta\Delta_\zeta J)\Delta\hat{q} = \Delta\tilde{q}, \tag{28}$$

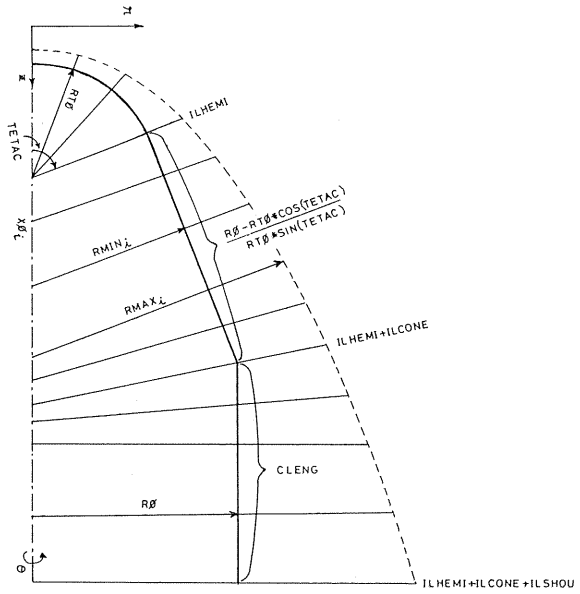

Fig. 1. The Body Geometry Used in HYREFS.

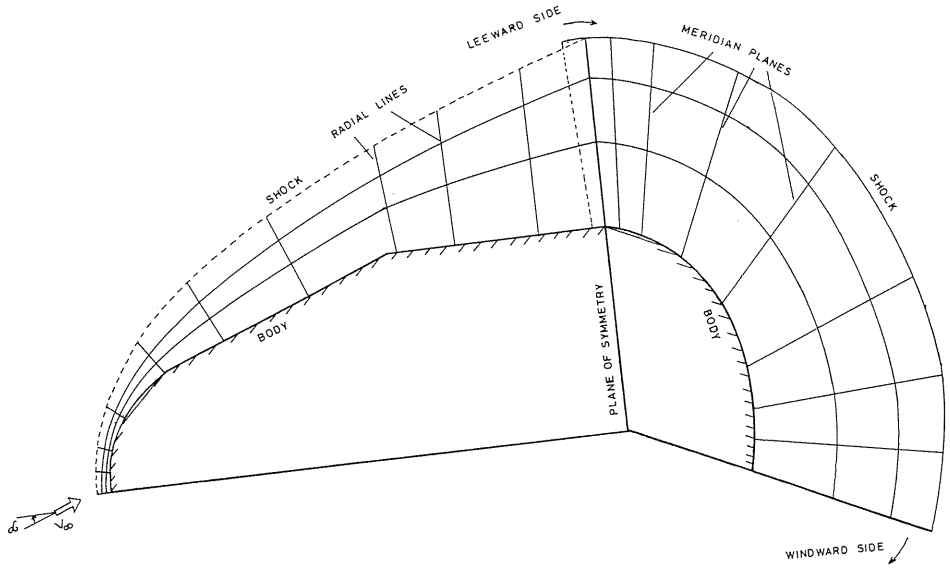

Fig. 2. The Physical Flow Plane.

and the solution is updated using

$$\hat{q}^{n+1} = \hat{q}^{n} + \Delta\hat{q} \,. \tag{29}$$

Here δ's are central-difference operators, Δ's are forward-difference operators, and ∇'s are backward-difference operators. The following difference schemes are employed :

$$\delta_{\xi}(\ ) = \frac{(\ )_{i+1,j,k} - (\ )_{i-1,j,k}}{2\Delta\xi}\ ; \quad \delta_{\eta}(\ ) = \frac{(\ )_{i,j+1,k} - (\ )_{i,j-1,k}}{2\Delta\eta}\ ;$$

$$\xi_{\zeta}(\ ) = \frac{(\ )_{i,j,k+1} - (\ )_{i,j,k-1}}{2\Delta\zeta}\ ; \quad \Delta_{\xi}(\ ) = (\ )_{i+1,j,k,} - (\ )_{i,j,k}\ ;$$

$$\Delta_{\eta}(\ ) = (\ )_{i,j+1,k} - (\ )_{i,j,k}\ ; \quad \nabla_{\xi}(\ ) = (\ )_{i,j,k} - (\ )_{i-1,j,k,}\ ; \text{ etc.}$$

The Jacobian matrices in Eqs. (26) through (28) are defined as

$$\hat{A}^{n} = \frac{\partial\hat{E}^{n}}{\partial\hat{q}^{n}}, \quad \hat{B}^{n} = \frac{\partial\hat{F}^{n}}{\partial\hat{q}^{n}}, \quad \hat{C}^{n} = \frac{\partial\hat{G}^{n}}{\partial\hat{q}^{n}},$$

$$\hat{P}^{n} = \frac{\partial\hat{E}_{v}^{n}}{\partial\hat{q}^{n}}, \quad \hat{Q}^{n} = \frac{\partial\hat{F}_{v}^{n}}{\partial\hat{q}^{n}}, \quad \hat{R}^{n} = \frac{\partial\hat{G}_{v}^{n}}{\partial\hat{q}^{n}},$$

representing the time linearization of the respective fluxes. The elements of these matrices have been derived by going through a cumbersome differentiation and are accurately implemented in these codes. Since they rarely require modifications, we are avoiding presenting those tedious expressions in this manual. However, an interested user may refer to References 5 and 10 for the details.

The two constants ε_I and ε_E are the dissipation coefficients added to the implicit side and explicit side of the algorithm respectively. Dissipation in the explicit side is necessary to damp any higher-order frequencies present in the solution. A suitable value should be chosen for ε_E. Since the chemistry is very sensitive to this added diffusion, the value should be limited to as small as possible. A fraction $\frac{1}{800}$ is used in the HYREFS codes. Based on the stability point of view an equivalent dissipation is also to be added to the implicit portion of the algorithm. ε_I in these codes is set equal to twice that of ε_E. Finally, I is the unit-block matrix.

2. 4. Decoding

Eqs. (26) through (28) result into a system of block-tridiagonal matrices with $\Delta\hat{q}$ as unknown. Gauss-elimination method has been employed in these codes to solve this system of matrices for $\Delta\hat{q}$. Then Eq. (29) gives new conservational variables $\hat{q}$ at (n+1)-th time step. The physical variables ρ_i, u, v, w, p and T must be decoded from the conservational variables ρ_i, ρu, ρv, ρw and e. Variables ρ_i, u, v and w can be directly obtained but pressure and temperature have to be decoded iteratively.

By rearranging the equation of state (3), a relation for temperature can be expressed as

$$T = \frac{1}{\beta_1 \sum_{i=1}^{N_s} (\rho_i / M_i)} \left[\sum_{i=1}^{N_s} \rho_i h_i + \frac{1}{2}\rho(u^2 + v^2 + w^2) - e \right] . \tag{30}$$

Since ρ_i, u, v, w and e are known, the above equation can be written as

$$T - a \sum_{i=1}^{N_s} \rho_i h_i = b \ , \tag{31}$$

where a and b are known constants. Eq. (31) is having the only unknown T, since h_i can be expressed as functions of T. Therefore, the above equation can be put in functional form as

$$f(T) = T - a \sum_{i=1}^{N_s} \rho_i h_i - b \ , \tag{32}$$

By sing the polynomials (Eq. (16)) for h_i, the above Eq. (32) is solved for T using Newton-Raphson method.

2. 5. *Boundary Conditions*

The physical (Fig. 2) flow domain and hence the computational region (Fig. 3) are completely enclosed by the five boundaries, namely (1) body surface, (2) shock surface, (3) exit flow pane, (4) plane of symmetry and (5) the singular line. Since the governing equations (Eq. (1)) are hyperbolic with respect to time, they represent an initial-boundary-valued problem. Therefore, in order to solve this problem, we must know the initial solution (i.e. at t=0) and the values of dependent variables on the five boundaries for all times $t>0$. If we are looking for a time-dependent solution, the initial solution and the boundary values must be specified accurately. For a steady-state solution we need only an approximate initial solution and compatible boundary schemes to guess the boundary values. An incompatible boundary scheme may result in either a diverging solution or a non-physical solution. This section gives details of the boundary schemes employed in HYREFS codes.

The finite-diference forms of the governing Eqs. (26)—(29) are solved over the grid points that lie within these surfaces but not on them, and hence values of the flow variables at all interior points are known.

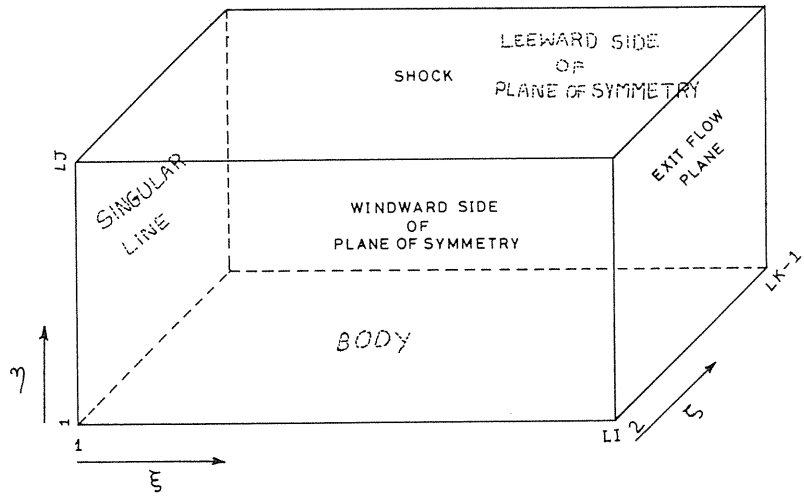

Fig. 3. The Computational Flow Plane.

2. 5. 1. Body surface

Density ρ is extapolated linearly at the grid points on the body surface from the nearby grid poins in the flowfield. The choice of body-oriented coordinate system makes the contravariant velocity component V equal to zero in invicid flows and U, V, W equal to zero in viscous flows. A normal momentum Eq. (11) with V equal to zero can be written as

$$\frac{\partial p}{\partial n} = (\eta_z \xi_z + \eta_r \xi_r + \frac{1}{r^2}\eta_\theta \xi_\theta) p_\xi + (\eta_z^2 + \eta_r^2 + \frac{1}{r^2}\eta_\theta^2) p_\eta$$

$$+ (\eta_z \zeta_z + \eta_r \zeta_r + \frac{1}{r^2}\eta_\theta \zeta_\theta) p_\zeta$$

$$= \rho [\, \partial_\tau \eta_t + u \partial_\tau \eta_z + v \partial_\tau \eta_r + w \partial_r (\frac{n_y}{r}) \,]$$

$$- \rho U (\eta_z u_\xi + \eta_r v_\xi + \frac{\eta_\theta}{r} w_\xi) - rW (\eta_z u_\zeta + \eta_r v_\zeta + \frac{\eta_\theta}{r} w_\zeta) . \qquad (33)$$

With the extrapolated values for U and W in invicid flows, the above equation will have only an unknown variable p for both viscous and inviscid flows. By adopting central differencing for p_ξ and p_ζ and upwind differencing for p_η, a finite-difference form of normal momentum equation can be written. Approximate factorization results in a system of tri-diagonal equations, which can be easily solved for pressures using Gauss elimination procedure.

For an adiabatic wall normal component of the temperature on the wall surface should be made equal to zero, i.e.

$$\frac{\partial T}{\partial n} = 0 . \qquad (34)$$

By following the solution procedure used to solve Eq. (33), the above equation is solved for wall temperature. If the wall is isothermal, its temperature is to be fixed at a specified value.

A non-catalytic wall yields zero outward normal gradients for the mass-fractions

$$\frac{\partial f_i}{\partial n} = 0 , \qquad (35)$$

which is solved for f_i similar to p. Wall catalysis acts in such a way that the active radical molecules near the wall return back to their inactive freestream states and a fully-catalytic wall does the job completely : Only fully-catalytic walls are considered in this study. Mathematically this can be written as

$$f_i = f_{i\infty} , \quad i = 1, N_s . \qquad (36)$$

The remaining unknowns on the wall can be determined by solving state equations using the known ρ_t, p and T.

2.5.2. Shock surface

In any supersonic flow around a blunt body the disturbed flow will be bounded by a peripheral shock. It's well known that at hypersonic speeds the bow shock will no longer become sharp but will have a thickness equal to several mean free paths. Present numerical scheme can capture this bow shock but it will be diffused much severely than the real shock waves and initiate chemical reactions within the shock wave. To avoid such reactions within the peripheral shock it is treated as a boundary and "frozen shock wave" conditions[12] are explicitly imposed across that. Peripheral shock is always made to coincide with the outermost grid surface. As the shock wave moves toward a steady-state location, the whole grid system shrinks or expands accordingly. The logic and procedure[13] involved are outlined here.

Rankine-Hugoniot relations for a moving discontinuity can be written as

$$h_2 = h_\infty + 0.5(p_2 - p_\infty)\left(\frac{1}{\rho_2} + \frac{1}{\rho_\infty}\right), \tag{37}$$

$$\lambda_{shock} = \left[(p_2 - p_\infty)\left(\frac{1}{\rho_\infty} - \frac{1}{\rho_2}\right)\right]^{\frac{1}{2}} - V_{n\infty}, \tag{38}$$

where λ_{shock} is the shock velocity in the outward normal direction and V_n is the flow velocity component normal to the shock wave. The subscript 2 indicates downstream shock properties whereas ∞ represents upstream freestream values. Frozen shock conditions are explicitly imposed by allowing no reactions across the shock wave. Hence, densities ρ_i beind the shock wave are calculated from the relation

$$\rho_{i2} = \frac{\rho_{i2}}{\rho_{2\infty}}\rho_{2\infty}. \tag{39}$$

After integrating the finite-difference equations over one time step Δt, pressures downstream of the shock wave at all grid points are extrapolated from the known field point values. Using the relations Eq. (37) through Eq. (39) along with the state Eq. (4), shock velocities can be calculated. Then the shock is moved by a distance equal to $\Delta t\,\lambda_{shock}$ at all the grid points on the shock surface. Based on the new shock locations grid surfaces are re-adjusted along all the fixed radial rays and the new shock geometry is evolved. Now, for the known shock geometry, all the flow variables downstream of shock can be obtained by solving Rankine-Hugoniot relations iteratively. These calculations start with an assumed value for density ρ_s. With the shock angle given, h_s and p_s are obtained from Eqs. (37) and (38), respectively. Temperature T_s corresponding to this h_s is determined by finding the root for the following equation:

$$g(T) = h_s - \frac{\rho_1}{\rho}h_1(T) - \frac{\rho_2}{\rho}h_2(T). \tag{40}$$

Further details on the calculation procedures are given in the References 10 and 14.

2.5.3. Exit flow plane

Though supersonic flows over blunt bodies will have both subsonic and supersonic flow regimes, the subsonic flow regime is limited to a pocket region around stagnation streamline.

Since our exit flow planes are always located far away from this pocket, flow in the neighborhood of this plane can be considered as supersonic. This situation leads us to a simple boundary condition that flow on this plane can be conveniently determined from the upstream flow values. Such a boundary condition can be justified even in the boundary layer region since a laminar boundary layer flow is governed by parabolic system of equations. Therefore, in HYREFS codes, flow variables on the exit plane are linearly extrapolated from the inner field points.

2. 5. 4. Plane of symmetry

Only a symmetric body configuration is considered. Flow over such bodies up to a moderate degree in angle of attack will not produce turbulance. Hence, there exists a plane about which flow will be symmetric ; this plane passes through the freestream velocity vector and the axis of symmetry of the body. Therefore, only one half of the flow is computed even though this is not a limitation to the procedure. Grid system in the meridian direction is constructed in such a way that one grid surface both in the windward side and leeward side should fall on the other side of the plane of symmetry. This means the grid surfaces immediately behind the boundary surfaces coincide with the plane of symmetry. With this, flow variables on the end meridian plane are calculated by utilizing the image principle. All the scalar variables such as pressure, density etc. are calculated from a quadratic extrapolation and by imposing maxima for these variables at the plane of symmetry. The vector quantity, e.g. velocity, is obtained by treating the plane of symmetry as a mirror.

2. 5. 5. Singular line

The choice of body-oriented coordinate system generates a singular line within the flow domain. The grid surface shrinks into a line at ξ equal to zero. In axisymmetric flows this singular line coincides with the stagnation streamline ; hence, no mass can flow across it. This condition is imposed by forcing radial velocity v and the contravariant velocity V to zero. Other variables are obtained by following the image principle as discussed previously. When the angle of attack deviates from zero, apparently there is no physical condition that can be satisfied along the singular line. Several methods have been tried before making this version of HYREFS codes. Two techniques are recommended and implemented in these codes. In the first one, flow variables are calculated by linearly extrapolating the values on the leeward side of plane of symmetry and the same values are used on all the other virtual planes at ξ=0. This works well for the flows up to 8 degrees of angle of attack. For higher angles of attack, errors arising from extrapolation grow faster and eventually terminate the calculation. Under these situations the second method is recommended. In this the singular line is completely avoided by starting the grid system from $\delta=\Delta\delta$ (Fig. 2). All the planes are then treated independently and the flow variables are simply extrapolated from the values on its corresponding plane. This leaves some error around the singular line but, nevertheless, reasonably accurate computations can be given up to 16 degrees of angle of attack. Details regarding how to choose either option will be given in the Input-Output section.

2. 6. Initial Conditions

In order to run these programs initial flow fields are necessary. As the governing equations are integrated using an implicit scheme, strictly speaking, any initial flow should converge to the final steady-state solution with same efforts (or the same number of iterations). When it comes to the real jugglery for the numerical values, one may add an adjective "compatible" to the subject "initial flow" in the previous sentence. Therefore, to have successful calculations, we need a compatible but not an approximate initial flow. The word "com-

patible" is somewhat tricky to be explained but it will be well understood with experience. However, to give some idea, it can be defined as any practical solution ; it doesn't matter to which problem it is applied. Sometimes, for doing a Mach 15 freestream flow calculation, a Mach 10 solution may be more compatible than an approximate solution developed for Mach 15 using Newton's method. Fortunately, at lower freestream Mach numbers, an approximate solution becomes compatible too. Therefore, if one is interested in simulating Mach 25 flows, it will be a reliable way to begin first with simulating converged solution for a Mach 5 flow starting from an approximate initial solution and then to gradually increase Mach number to 25, maybe in five steps.

The code HYREFS-NR2 is equipped with self starter routines. This code can give axi-symmetric solutions, without asking for initial flowfields, for the freestream Mach number range 10 and 30 and for the altitudes between 40 and 80 km. These solutions can then be used as initial flows for simulating three-dimensional reacting flows. Therefore, we can say these codes are "self-equipped", though a good management technique will be needed by the user. Similarly, solution over long bodies can be generated by gradually increasing the body length. Some automatic features are available in HYREFS for doing this, which will be explained later in the Input-Output section.

2. 7. Stability

If the chemical production terms $\dot{W}_i$ were identically equal to zero, then a linear stability analysis suggests

$$\Delta\tilde{\tau} = CFL\ min\left\{\frac{\Delta\xi}{\Lambda_{max}}, \frac{\Delta\eta}{\gamma_{max}}, \frac{\Delta\zeta}{\Phi_{max}}\right\}, \tag{41}$$

where Λ_{max}, γ_{max} and Φ_{max} are the eigenvalues of the matrices $\hat{A}^n$, $\hat{B}^n$ and $\hat{C}^n$ respectively. Being implicit in nature, CFL can take a high number. The presence of chemical production terms severely restricts this number to low values and it depends on the freestream Mach number. Local time step concept is used in HYREFS codes to accelerate convergence. This technique advances the flow for a locally determined time step. However, at any grid point, the local time step is not allowed to take a value grater than the specified value. HYREFS codes can choose a most appropriate CFL number for the given flow environment. A user can also set a specific CFL number by masking this operation.

3. Code Structure

Previous sections have detailed the methodology for generating hypersonic reacting flows. Actual implementation of it in a computer code using FORTRAN-77 language is described in this chapter. Though every effort has been made to give a clear picture about the HYREFS codes, it becomes impossible to cover all aspects. Comment cards inserted in the codes may fill this void. This section should be read in conjunction with the codes.

3. 1. Flow Chart

The overall flow diagram for the calculations is shown in Fig. 4. After acquiring all the required input data in the main program, calculations are initiated in subroutine FLOWAD. At first this subroutine calls READFF or READTD for reading either two-dimensional or

three-dimensional data. Then control goes to the subroutine PRESET where the body geometry and the mesh system will be obtained. Read data will be interpolated to this mesh by INTERP. Iterations over the initial data begin at GRID which is called from FLOWAD. GRID usually alignes the outermost grid surface with the shock wave and calculates immediate downstream flow variables.

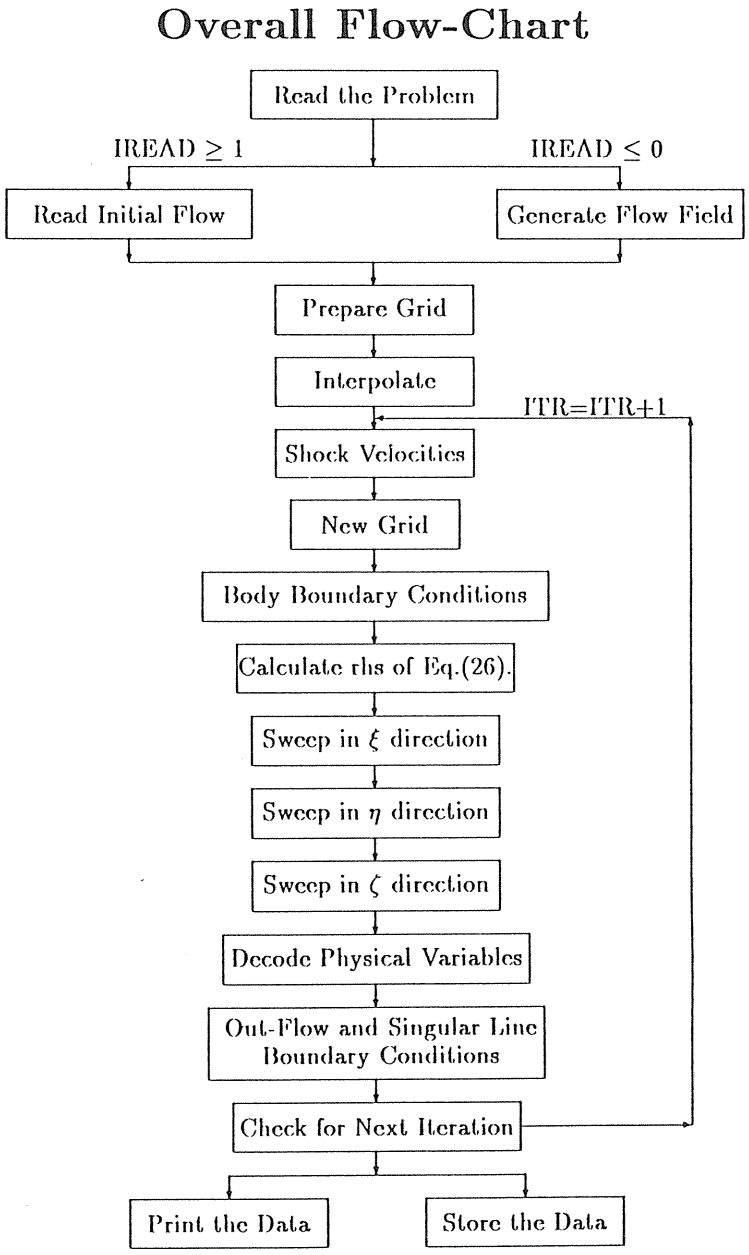

Fig. 4. The Basic Sequences Involved in Simulating a Chemically Reacting Flow.

3. 2. *Subroutine FLOWAD*

This is the central subroutine and calls all the other subroutines at an appropriate time. Values for the coefficients that appear in chemical kinetics (Eqs. (6)—(9)) and transport coefficients (Eqs. (17)—(23)) are supplied here using DATA statements. FLOWAD first calculates and prints out the freestream properties. The first subroutine called by this is either READAX or READTD for reading either axisymmetric or three-dimensional initial flow data respectively. In axisymmetric HYREFS codes the subroutine DOSELF generates initial flowfield by itself. All the control variables are defined and the body geometry and hence the grid system are prepared by calling PRESET. The read data will then be interpolated on to this new grid by INTERP subroutine. After making these preliminary preparations the finite-difference form of governing equations (Eqs. (26) to (29)) are solved repeatedly in the loop "DO 100 ITR=1, ITEND" up to ITEND times. This procedure first initiates SHOCK subroutine, where a peripheral shock is moved by a distance corresponding to the shock velocity and then, in the GRID subroutine, a new grid system is constructed by aligning the outermost grid surface with the moved shock wave. Wall boundary conditions are enforced in BODYBC. In the next few statements dissipation terms of right-hand side of Eq. (26) are estimated. It is followed by the calculations for transport properties and derivatives of physical properties such as p, T, f_i etc. RHS of Eq. (26) is then completed by calculating the inviscid, viscous and species production fluxes. The amount of change that should be added to the $\hat{q}$ vector is then determined by sweeping RHS of Eq. (26) in all the space coordinate directions. These operations are done in FSCAN subroutine. The argument value governs the direction of sweep. $\hat{q}$ vector is then updated by adding $\Delta\hat{q}$. Total residual left in the solution is calculated by summing up $\frac{\Delta_{\rho 2}}{\Delta_\tau}$ values at all grid points and then is printed out after every IPRES iteration. Physical flow variables are decoded (Fig. 5) from $\hat{q}$ by solving Eqs. (30) to (32). Since enthalpy and specific heat of one chemical species are described by a different polynomial in each of the five temperature ranges, decoding procedure turns out to be slightly complicated. Within one temperature range, solution for Eq. (30) is obtained by using Newton-Raphson technique. Cycle for one iteration is finally completed by calling OUTCND where flow variables on the out-flow boundary and on singular line (fictitious surface) are estimated. When the iteration count reaches ITEND or if the residual drops below a specified value or when the calculation time exceeds the allotted time, the solution goes to some peripheral subroutines such as FSAVE and PRNTFF.

3. 3. *Subroutine FDERV*

While calculating the Jacobian matrices $\hat{A}^n$, $\hat{P}^n$ etc., some derivatives such as $\tilde{\partial}_\xi\left[\frac{T}{\rho(\bar{R}-c_p)}\right]$ are repeatedly used. Therefore, such complicated derivatives are calculated once and stored here in FDERV subroutine. Details about these derivatives may be found in Reference 5. And also the grid Jacobians (ξ_z, ξ_r, ξ_θ etc.) are transferred into general arrays. This is necessary to have only one subroutine FSCAN for scanning in three different directions.

3. 4. *Subroutine FSCAN*

FSCAN has an argument NDIR which tells the subroutine in which direction it has to sweep the error matrix Δq. If NDIR is equals to 1, then scanning will be done in ξ direction. Similarly, the value 2 or 3 indicates either η or ζ direction. The finite-difference equations Eqs. (26) through (28) in fact represent an algebraic system of block diagonal matrices. Blocks in this system are square matrices with sizes depending on the number of equations that the governing equation Eq. (1) represents. A careful insight into these BDM (Block Diagonal Matrices) suggests that every lower diagonal block matrix has a similar upper diagonal

Flow-Chart for Decoding

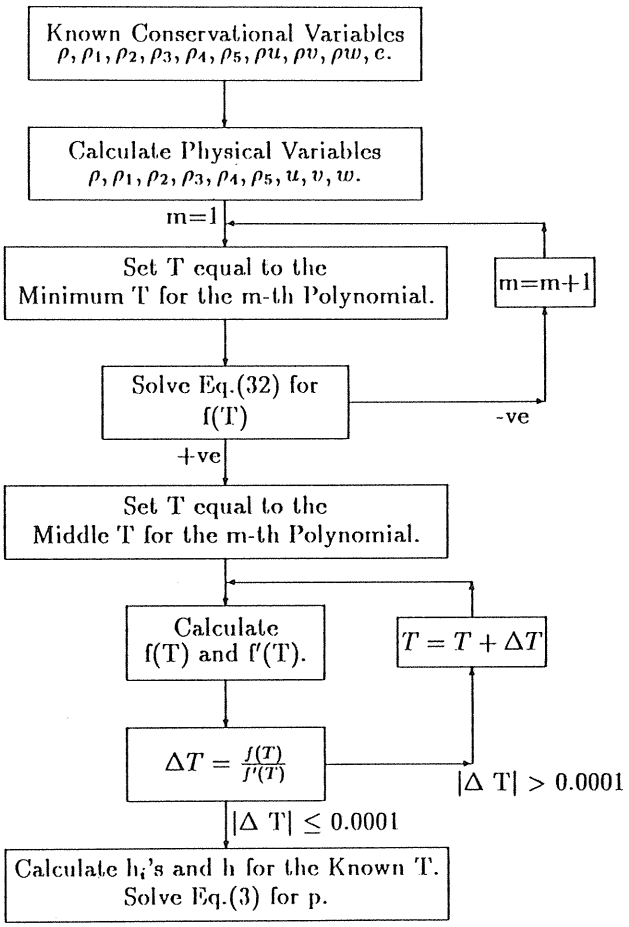

Fig. 5. Branching for Various Wall Boundary Conditions Employed in HYREFS Codes.

block matrix with only differences in diagonal elements. And also, these differences are only due to the grid parameters. Therefore, only upper diagonal block matrices are generated in FSCAN subroutine and stored in the array labeled 'AMU'. The differences in diagonal elements are stored in 'AML'. Diagonal blocks of BDM are truly diagonal and the diagonal elements are simple constants. They are stored in 'AMD'.

Sweeping operations or, in other words, solution of BDM should be carried out along a grid line and on all grid lines. To improve the efficiency of these calculations, all grid lines are stacked one over another and sweeping operations are performed by treating the whole lot of

stacked blocks as one unit. Of cource, FSCAN subroutine just arranges the blocks in this way and actual solution procedure is carried out by BDMSLV subroutine. Further improvement in efficiency in calculating the elements is obtained by merging sweep lengths and column specification into the stack heights. Therefore, AMU array has only two dimensions, first one representing a number equal to block size multiplied by total number of grid points and second one representing the block size. For example, in a three-dimensional five species problem, dimensions for various arrays are as follows:

Let LI - No. of grid points in ξ direction,
LJ - No. of grid points in η direction,
LK - No. of grid points in ζ direction,
LL=9 (No. of equations represented by Eq. (1).
Then AMU array will have (LJ-2)*(LK-2)*(LI-2)*9 $\times$ 9 elements, while AMR, AML and AMD will have (LJ-2)*(LK-2)*(LI-2)*9 elements.

The one block of (LI-2)*9 $\times$ 9 elements of AMU represents (LI-2) blocks of 9 $\times$ 9 matrices corresponding to a grid line on which sweeping will be performed. (LJ-2)*(LK-2) represents the total number of such grid lines. Right-hand side matrices are rearranged for such stacked operations and stored in 'AMR' array. Initiation of BDMSLV subroutine solves BDM and brings back the solution in AMR array. Therefore, finally, FSCAN subroutine puts back the trickily-arranged solution in AMR into the conventional manner in RHSM array.

3. 5. Subroutine BDMSLV

Argument NS gives the length of the sweep and NN1 the number of grid lines multiplied by the block size. Remember that these values change with the direction of sweep and depends on just grid size and block size. To save memory size, the temporarily vacant common block 'CB9' is made use of for solving BDM. For the usual grids, i.e. LI and LJ > LK, dimensioning used for the variables in 'CB9' will be normal. But for any reason if LK is chosen greater than either LI or LJ, then a minor editing is necessary at this subroutine. The variable LP7 in PARAMETER listing which is defined as 'LP7=(LI-2)*(LJ=2)' should be changed to the product of the largest grid parameters. For example, if LK is greater than LJ but still less than LI, then it should be 'LP7=(LI-2)*(LK-2)'.

BDMSLV uses Gauss elimination procedure for solving BDM but all operations have a specific vector length. Explanation for this routine is skipped with the fear that any such attempt may cause confusion. However, an interested reader can learn by himself by working with the code. Fundamental knowledge on computer architecture is of cource essential.

Control variable 'CONDK' in three-dimensional codes is introduced to implement symmetry conditions while making sweeps in ζ direction. This assures perfectly axisymmetric solution when angle of attack is set equal to zero.

3. 6. Subroutine BODYBC

Surface boundary conditions are enforced in this subroutine. Flow diagram for this subroutine is shown in Fig. 6. Wall state is completely defined with the four variables that appear in the argument list. Meanings of the various levels in these arguments are as follows:

IVCOND $\leq$0 - Flow is viscous,
IVCOND $\geq$1 - Flow is inviscid,
ITCOND $\leq$1 - Wall is adiabatic,
ITCOND $\geq$1 - Wall is isothermal,
ISCOND $\leq$1 - Wall is completely non-catalytic,
ISCOND $\geq$1 - Wall is fully-catalytic,
TWALL - Wall temperature to be specified for isothermal walls.

Flow-Chart for Wall Treatment

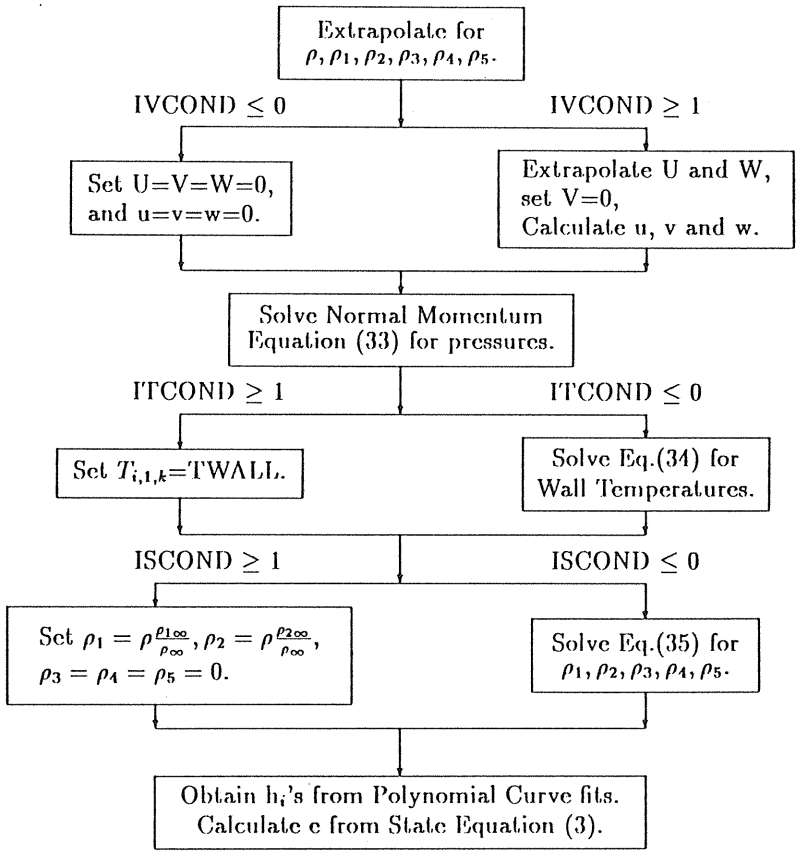

Fig. 6. Calculation Procedure for Obtaining Physical Flow Variables from Conservational Variables.

As discussed in the subsection 2.5, some conditions of the above list require enforcement of zero normal derivatives of a corresponding variable. Other conditions require specific values for those variables. Pressure is always calculated by solving normal momentum Eq. (33). Variable 'JVAR' is set with a number that is equal to the number of variables that require zero normal derivatives. The tri-diagonal system of equations are solved using Gauss elimination procedure. Finally, energy, enthalpy and specific heats are calculated on body surface at a given temperature.

3. 7. Subroutine PRNTFF

This peripheral subroutine prints out the numerical data whenever is called. The variable that is to be printed out can be selected by choosing a nonzero value for the corresponding

variable in the argument list. First, shock and body distances along all the radial rays that lie in a maximum of 18 meridian planes will be printed out. Then the numerical data of the specific flow variable printed in five sets, one for each meridian plane. 16 points will be chosen along ξ direction, while all points in η direction will be covered. More data can be printed by editing the code.

3. 8. Subroutine FFSAVE

This subroutine can be called for saving the necessary data. The saved data are sufficient for restarting the calculation at a later time. Usually, it will be activated after successful completion of calculation. Values stored under the variable 'IGAS' will act as a key to open necessary loops while reading the data. More discussions on this important variable will be postponed till Input-Output section.

3. 9. Subroutine READAX

Two-dimensional data will be read here. From the first read card it determines the nature of the following data. As a saftey measure to prevent reading from unwanted data file, the first read value will be compared with the asked value. If both don't match, computations will automatically be terminated. By assigning a value to the variable 'IREAD' one indicates the nature of the data stored in that file. That is whether a two-dimensional or three-dimensional, a reacting flow or frozen chemistry etc.

But it allows variations in grid systems. If the stored data are for a grid system that is smaller than the one used in the present calculation, then the code automatically extends the read data on to this larger grid. Similarly, it tailors the larger data to fit a smaller grid. This capability can be best used for increasing the body length gradually. However, the number of grid points in η direction should not be changed, which of cource seldom happens. Another important point to remember is that the values of 'IREAD' and 'IGAS' used for a calculation can be different, and as a matter of fact IREAD has nothing to do with the calculation requirements. If IREAD differs from the calculation requirements, then this subroutine automatically developes required flowfield from the read data. This capability can be used for doing three-dimensional calculations by reading 2-d data or for doing reacting simulations by reading frozen chemistry flowfield, etc.

3. 10. Subroutine READTD

Three-dimensional data will be read from this subroutine. The features on this subroutine are almost similar to those of READAX. While using the ability to extend a grid system it should be made certain that grid surfaces in η as well as in ζ direction are not changed.

3. 11. Subroutine OUTCND

This is the final subroutine called in one iteration for the flowfield. Values for the flow variables on both singular and out-flow surfaces are estimated here. A linear extrapolation is used. When there are severe changes in flow variables, a first-order linear extrapolation may be appropriate. However, an extrapolation of order of accuracy between zero and first but close to zero-th seems to be more stable when the variations are not severe. The order of accuracy for extrapolation can be fixed by assigning a value to the variable 'AL1'; zero for the zeroth-order and one for the first-order. For the out-flow boundary calculation it is set equal to a value close to 0.2 and for the singular surface calculation it is about 0.9. In order not to tighten the system, pressure is relaxed to take any errors that may be raised due to extrapola-

tion. Additional checks for non-physical data are also incorporated. Plane-of-symmetry conditions are reenforced on these two surfaces.

3. 12. Subroutine PRESET

This subroutine does all the preliminary calculations. It fixes flags for the control variables. These control variables are used to skip regular calculations on boundary grids. For example, outer two layers should be skipped while doing fourth-order dissipation calculations and the outermost layer should be skipped while calculating second-order derivatives. Of course, calculations on these skipped layers will be done in a special way. Usage of such control variables with flag identifications is proved to be an efficient way to mask "do loop" operations on vector machines. This subroutine constructs the body surface and calculates the nondimensional grid spacings along radial lines which are identical on all radial lines. If the program is directed to generate initial data by itself, then the subroutine DFRMAX will be activated at this point. This supplies an approximate peripheral shock geometry. Otherwise, the shock geometry will be obtained from the read data. Once the shock geometry is known, this PRESET subroutine constructs the grid system and calls for the metric coefficients.

3. 13. Subroutine DFRMAX

If the program is instructed to generate initial data by itself, then this subroutine will be called upon. Here an approximate shock shape will be determined. If any problem is seen in completing calculations with this shock geometry, a user can supply his guess for the shock shape here. Care should be taken while preparing such data. RMAXs are the radial distances on radial rays which may not have a common center.

3. 14. Subroutine INFLOW

This should be used in conjunction with DFRMAX. Once the shock geometry is fixed at DFRMAX, a grid system will be constructed at PRESET. At this point INFLOW generates a flowfield on this grid based on the freestream properties. Newton's method is employed. With these data one can generate solutions for perfect or frozen gas, viscous or inviscid gas models for a wide range of Mach numbers and altitudes.

3. 15. Subroutine GRID

This subroutine always constructs a new grid system for the latest shock geometry. And furthermore it calculates the downstream flow vaiables immediately behind the shock wave. This is possible when the shock geometry is known. The calculation sequences are depicted in Fig. 7. First, it estimates the shock angle. Then the modified Rankine-Hugoniot relations (Eqs. (37) and (38)) are solved using Newton-Raphson iteration procedure. Some check criteria are provided on the number of iterations and also on the accuracy for convergence. During the calculations, warning messages will be issued whenever there is a problem in obtaining a valid downstream flow for the given shock angle. However, solution procedure will be continued with these erroneous flows expecting that such errors will disappear after a few iterations for the solution. If these errors magnify and spread into the flow domain, then the calculations will be terminated abruptly in FLOWAD subroutine. A study on these warning messages may help in identifying the problem. Again, symmetry conditions are reenforced on this shock surface.

3. 16. Subroutine METRIC

This routine calculates metric coefficients for the given grid system and for the used coordinate transformation. Central differencing is used.

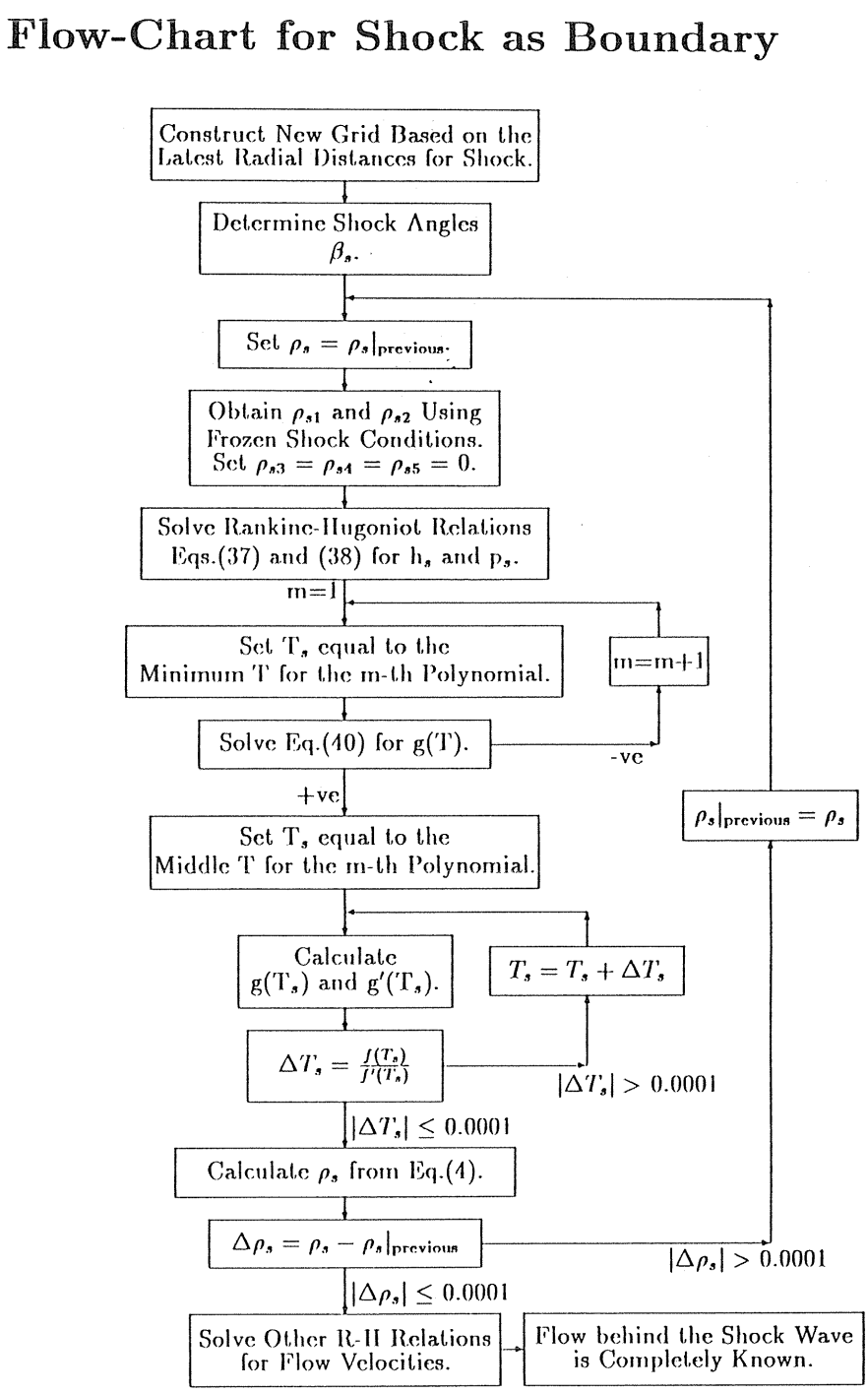

Fig. 7. Procedural Steps Involved in Estimating Shock Velocities.

3. 17. Subroutine INTERP

It interpolates read data on to the grid system with which one starts calculations. Remember that this subroutine will be called upon only once after PRESET. Flow variables need not be interpolated for the subsequent changes in the grid system. This is perfectly taken care of by the algorithm. INTERP also serves to fill in the vacant arrays that are necessary for starting calculations. Since only the absolutely necessary flow variables are stored in the data storage files, other flow variables are to be calculated from these stored variables in order to prepare data for a fresh calculation. This will be done here in this subroutine.

3. 18. Subroutine SHOCK

Shock geometry is modified here. This can be considered as a feedback subroutine. Solution for the flowfield starts with a peripheral shock geometry. When this flowfield does not match with the shock wave, perturbations over flow variables will appear. Since we are solving correct governing equations for a physical problem, such perturbations will always try to oppose the errors. Therefore if we adjust the wrong shock wave with a guidance obtained from these perturbations, then we will most likely have a proper feedback. In this subroutine, perturbations in pressure are tapped out and used to determine the shock velocities. Pressures behind the shock wave are estimated by extrapolating pressures in the flow domain. Feedback is restricted by correcting only a fraction of pressure behind the shock wave. Amount of feedback should match with the error propagation. The values fixed in these codes are assumed to be the best. Shock velocities are calculated from these modified pressures by solving again Rankine-Hugoniot relations. Methodology for estimating shock velocities is shown in Fig. 8. Then shock wave is moved with these velocities at all grid points on shock boundary. Since the whole grid system either expands or shrinks accordingly, grid velocities at all grid points are calculated proportionally so as to keep such velocities on the body surface at zero.

3. 19. BLOCK DATA

Values for the coefficients used in the enthalpy and specific heat polynomials are put into the common blocks in this subroutine. These values are also listed in Tables 2 through 4.

4. Input-Output

This section explains how to use HYREFS codes. While all these codes except the graphics versions require eight input-cards to perform a calculation, graphics codes require three to four. Instructions for preparing these input-cards are given in the following subsections. The first eight cards are common for HYREFS-NR2, HYREFS-CR2, HYREFS-NR3 and HYREFS-CR3. Last subsection describes how to prepare input-cards for HYREFS-GR2 and HYREFS-GR3.

4. 1. Editing

Before preparing data for a calculation, a user has to edit the present code to satisfy his own particular grid size. The basic variables that define grid size are LI, LJ and LK. In two-dimensional codes there is no LK. All these variables are assigned with values in PARAMETER listing of each subroutine and they always appear in group. Editing can be done quite easily from a terminal station. The axisymmetric HYREFS are made for a 30×30 grid

Flow-Chart for Shock Velocities

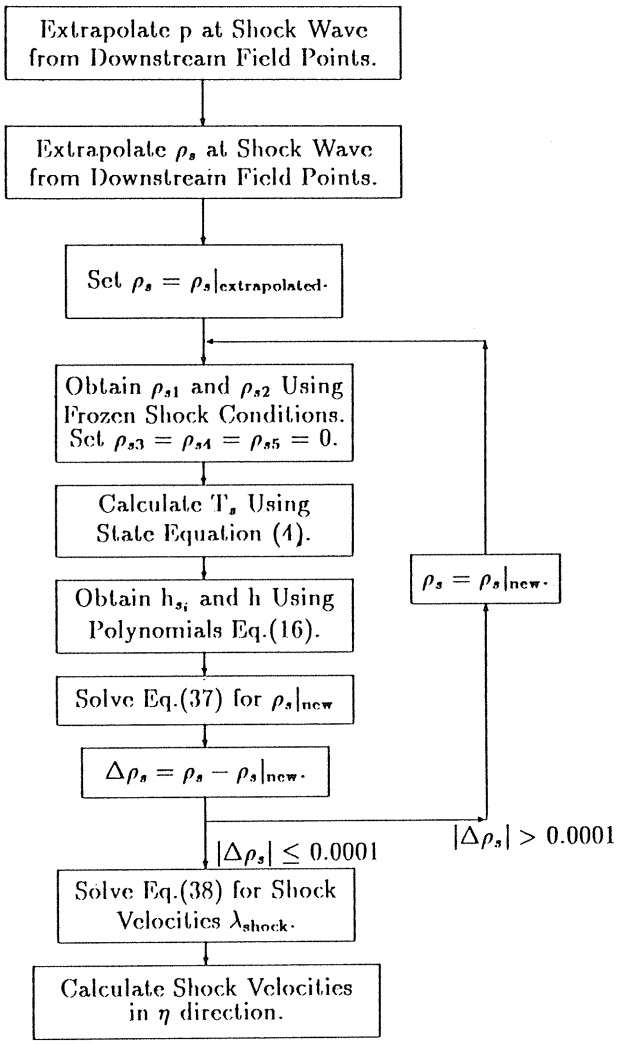

Fig. 8. Determining Flow Properties behind a Known Shock Wave.

system and the three-dimensional ones are for 30 × 30 × 20 grids. Therefore, a user who wants to perform calculations on a 60 × 35 grid system should change all the groups of letters 'LI=30, LJ=30' into 'LI=60, LJ=35'. Such groups nowhere else occur and hence even a blind conversion using a TSS command 'C ... ... ALL' is quite safe. Similarly, to set the 3-d codes for a 60 × 35 × 20 grid system, all the groups 'LI=30, LJ=30, LK=20' should be

changed to 'LI=60, LJ=35, LK=20'.

If for any reason LJ is preferred to be less than LI in a 2-d code, the value of LP7 in PARAMETER listing of BDMSLV subroutine should be changed from 'LP7=(LI-2)*LL' to 'LP7=(LJ-2)*LL'. Similarly, in a 3-d code, LP7 should be determined from the two largest values among LI, LJ and LK. In the codes this is set LI and LJ.

4. 2. Input Body

Card number 1

The first data card to be supplied is for describing body geometry. The basic configuration used is Hemisphere-Cone-Cylinder. By choosing proper lengths for each section it can be modified to a Hemisphere-Cone or Hemisphere-Cylinder. This card should contain seven values. The variables reading these data are RT0, R0, TETAC, CLENG, ILHEMI, ILCONE and ILSHOU whose meanings are as follows:

RT0 ; Tip or hemisphere radius in meters.
R0 ; Shoulder or cylinder radius in meters.
TETAC ; Half angle of the cone in degrees.
CLENG ; Length of the cylinder or shoulder in terms of RT0.
ILHEMI ; Number of grid points in ξ direction that should lie on hemisphere.
ILCONE ; Number of grid points in ξ direction that should lie on conical portion.
ILSHOU ; Number of grid points in ξ direction that should lie on shoulder or cylinder portion.

Example 1 : For a hemisphere-cone-cylinder body ;
1.0, 1.8, 20.0, 2.0, 15, 30, 15 [card 1]
Example 2 : For a hemisphere-cylinder body ;
1.0, 1.0, 00.0, 5.0, 20, 00, 40 [card 1]
Example 3 : For a hemisphere-cone body ;
1.0, 2.5, 30.0, 0.0, 14, 46, 00 [card 1]

It is important to remember that the total ILHEMI+ILCONE+ILSHOU may be greater than or equal to but never be less than LI. If it is greater than LI, then only the portion of body up to LI will be taken from the defined body length. If one wants to simulate flows over long bodies, then the body geometry can be defined once for all and calculations can be started from a small portion of the body by choosing small value for LI and then repeat the calculation by gradually increasing the body length by increasing just LI. This is based on the same grid distribution over the portion on which calculations were already made.

4. 3. Input Freestream

Card number 2

Flight conditions are identified by (i) pressure and density at the flying altitude and (ii) Mach number to represent flight speed. Though flight speed is a more familiar input data, Mach number is used for convenience. The corresponding flight speed will be printed out in the output list. Besides these parameters, it is necessary to feed values for angle of attack, molar percentage of oxygen molecules in the freestream. The variables read here in sequence are AMACH, ALPHA, ALTIT, PFRE, RFRE and OXYMP. Meanings and units of these variables are as follows:

AMACH ; Flight Mach number.
ALPHA ; Angle of attack in degrees.
ALTIT ; Flying altitude in kilometers.
PFRE ; Atmospheric pressure at this altitude in $\frac{N}{m^2}$.
RFRE ; Atmospheric density at this altitude in $\frac{kg}{m^3}$.
OXYMP ; Molar percentage of oxygen in the freestream.

Examples of this card are

Sample 1 : For a Mach 15 flight at 70 km altitude ;

15.0, 5.0, 70.0, 5.5205, 8.7535 D-05, 20.0 [card 2]

Sample 2 : For a Mach 30 flight at 80 km altitude ;

30.0, 0.0, 80.0, 1.0366, 4.3350 D-05, 20.0 [card 2]

4. 4. Input Gas Model

Card number 3

Description about the gas model (used in the calculation) should be done here by putting a tag value to the variable 'IGAS'. The following notation has been employed :

IGAS=1 ; two-dimensional perfect gas model.
IGAS=2 ; two-dimensional frozen chemistry gas model.
IGAS=3 ; two-dimensional chemically reacting five species gas.
IGAS=11 ; three-dimensional perfect gas model.
IGAS=12 ; three-dimensional frozen chemistry gas model.
IGAS=13 ; three-dimensional chemically reacting five species gas.

Another variable that will be read along with IGAS from this card is IREAC. For a non-reacting flow it should be asigned with zero. Actually, this variable controls the progress of chemical reaction during a calculation. For Mach numbers above 20, it may be a good idea to allow chemical reactions to progress gradually with the increase in the number of iterations. While doing chemically reacting flow calculations, only a portion of the calculated species production rate (Eq. (12)) will be added to the RHS of Eq. (26) for all the iterations that are less than IREAC. When iteration number exceeds IREAC value, then all the limitations on chemical reactions will be lifted-off. Therefore, when the freestream Mach numer is less than 20, IREAC can be set equal to one. In any case, a maximum of 400 should be sufficient for higher Mach numbers. Example : For three-dimensional chemically reacting flow calculations ;

13,200 [card 3]

4. 5. Input Body Boundary Conditions

Card number 4

This card contains information on body surface properties. There are four variables to describe this :

IVCOND ; Differentiates viscous and inviscid calculations ;
−1 for viscous flow, +1 for inviscid flow.
ITCOND ; Differentiates adiabatic and isothermal walls ;
−1 for adiabatic wall, +1 for isothermal wall.

ISCOND ; Differentiates non-catalytic and fully-catalytic walls;
−1 for non-catalytic wall, +1 for fully-catalytic wall.

TWALL ; If an isothermal wall is used, then wall temperature should be supplied in degrees Kelvin.

One can choose any meaningful combination for the above variables. However, caution must be given to avoid wrong boundary conditions. Some physically implausible boundary conditions are:

- Isothermal wall under inviscid flow environment.
- Adiabatic and fully-catalytic wall.

Some frequently used cards are:

Example 1: For an adiabatic and non-catalytic wall in viscous flow;

−1, −1, −1, 0.0 [card 4]

Example 2: For an isothermal and non-catalytic wall at temperature 1200 K in viscous flow;

−1, +1, −1, 1200.0 [card 4]

4. 6. Input Grid Parameters

Card number 5

HYREFS codes use clustering in η direction. Clustering of grid points is necessary near the body to capture gradients within the boundary layer and near the shock wave to allow severe dissociation just downstream of it. A user may have to select a certain clustering to obtain satisfactory results. It is usually done by choosing relative values for the variables CLUSTB, CLUSTS and CLUSTM.

First a central surface will be drawn from the value of CLUSTM. If the total distance between the body and shock is unity, then CLUSTM represents the fraction of the distance between the body and central surface. For example, 'CLUSTM=0.5' means that the central surface is just in the middle of the body and shock surfaces. Total grid points in η direction are then divided into two halves. One half of these points lies between the body and central surface and the other half lies within the central surface and peripheral shock. Had the grid points been distributed uniformly in their respective regions, then one could figure out two spacings, one for each region. Now CLUSTB and CLUSTM indicate desired spacings near the body and shock relative to their respective uniform spacings. Based on these values grids will be stretched exponentially toward the central surface. The amplification factors on both stretchings are automatically determined from the criterion that at the central surface grid spacings calculated from both sides should agree. The ranges of values for these grid parameters are as follows:

$0.1 \leq$ CLUSTB ≤ 1.0 where 0.1 given highly clustered and 1.0 giving uniform grids near the body.

$0.1 \leq$ CLUSTS ≤ 1.0 where 0.1 giving highly clustered and 1.0 giving uniform grids near the shock.

$0.4 \leq$ CLUSTM ≤ 0.6 where 0.4 giving more grid points near the body and 0.6 giving that near the shock.

Example 1: For highly clustered grids;

0.2, 0.2, 0.5 [card 5]

Example 2: For relatively uniform grids;

0.7, 0.8, 0.4 [card 5]

4. 7. *Input About Initial Flowfield*

Card number 6

Generally speaking, HYREFS codes require an initial flowfield to start with. However, to make these codes self-sufficient, HYREFS-NR2 is equipped with self-starters. One can generate either a perfect gas solution or a frozen gas solution for a wide range of freestream conditions using this code straight-forwardly. These solutions are supposed to be sufficient to start other codes. Since there are a number of variations to the input flowfield, a user has to specify a tag value to the variable 'IREAD' to indicate what the input flow is. Notation employed to describe IREAD is identical to that used for IGAS, except for the additional value zero to ask the code to generate its own data. In other words, if IREAD is greater than zero, then it should be identical to the IGAS value of the supplied initial flowfield. Naturally, the option 'IREAD=0' can only be used in HYREFS-NR2 code. Since a user can seek solution for a particular gas model by supplying a completely different gas model solution, IREAD needs not to be the same as IGAS of the present calculation.

Example 1 : To simulate a three-dimensional reacting flow from the supplied axisymmetric frozen gas solution ;

2 [card 6]

Example 2 : To simulate a three-dimensional reacting flow from the supplied three-dimensional reacting gas solution ;

13 [card 6]

4. 8. *Input About Calculations*

Card number 7

Calculations for the current run will be terminated 'normally', if any of the following conditions is satisfied :

- If the number of iterations on solution exceeds the specified value of 'ITEND'.
- When total cpu time of the calculations reaches the maximum allowed time in minutes which is specified with the variable 'ITIME'.
- If the residual of the solution falls below a specific value input through the variable 'STEADY'.
- Finally, when IECODE becomes non-zero.

Values for the variables ITEND, ITIME and STEADY should be supplied here in Card 7 along with CFLNO and ISING. All the codes can select an appropriate CFL number for the calculation depending on the flow conditions. However, this feature can be masked if one wants to calculate using a particular CFL number. A non-zero positive number assigned to CFLNO will perform the calculation with that number as CFL number. On the other hand, if CFLNO$\leq$0.0, then the built-in logics will be reckoned to select the best CFL number. As discussed in the previous chapter, there is an option for treating the singular line as a boundary. Either of the two possible options can be selected by assigning a value to the variable ISING. For ISING=0, flow on the singular line is calculated at the plane of symmetry on the leeward side, whereas, in the other option, the singular line is eliminated by temporarily removing 'ISING' number of initial j=constant grid surfaces. After the calculation flow on the erased surfaces will be extrapolated.

Example 1 : For 2-d reacting flow calculations ;
4000, 12 , 1.0 D-10, 0.0, 0 [card 7]
Example 2 : For 3-d frozen flow calculations ;
3000, 30, 1.0 D-09, 0.0, 1 [card 7]
Example 3 : For 3-d reacting flow calculations ;
3000, 50, 1.0 D-09, 2.5, 0 [card 7]

4. 9. Input Data Management

Card number 8

The above 'normal' termination will finally go to some peripheral routines for storing and printing. It is advisable to store the data before termination ; however, a user may opt to store the data only when a particular kind of termination happens. Value to ISTORE gives such instructions. Tags for ISTORE are :

ISTORE=0 ; Data will not be stored.
ISTORE ≥ 1 ; Data will be stored.
ISTORE=1 ; If termination is 'normal' and if IECODE=0.
ISTORE=2 ; If termination is 'normal'.
ISTORE=3 ; Only if the residual falls below STEADY.
ISTORE=4 ; Only if the iteration count exceeds ITEND.

To keep track of convergence, the residual at regular intervals can be printed out along with the iteration count, time step and flow advancement time. This interval can be decided by assigning a value to the variable 'IPRES'. If it is zero, then no such printing will be done. If it has a value greater than zero, then residual printing will be done after every IPRES iteration. Finally, the numerical data of any physical variable can be printed at the end of calculation, if the termination is 'normal'. The variable for printing should be indicated by flagging on the counter for that variable. The counters are IPPS, IPDS, IPTP, IPVU, IPVV, IPVW, IPSP1, IPSP2, IPSP3, IPSP4 and IPSP5 for pressure, density, temperature, u-component of velocity, v-component of velocity, w-component of velocity and mass-fractions of the five species, respectively.

4. 10. Input For Graphics

Card numbers 1 to 4

Flow solutions can be explained more lucidly by plotting various graphs. One can imagine endless number of pictures from flow data, but it will be a mammoth task to include every possible picturization in a graphics code. Fortunately they can be limited to only some striking graphs based on individual's imagination. Two supporting graphics codes HYREFS-GR2 and HYREFS-GR3 are developed to draw some important graphs from two-dimensional and three-dimensional data, respectively.

Solutions from HYREFS codes are stored in a data-file in four lines. The first line contains informations on gas model, grid size etc. Freestream properties and the values for some constants can be read from the second line, while mesh description can be retrieved from the third line. All the necessary physical flow variables are stored in the last line. Immediately after reading the data from the first line, these codes will automatically be set for responding to subsequent data ; hence a user in fact needs not to have to remember about the stored data.

Two-Dimensional Graphs

HYREFS-GR2 can plot the following graphs:

- Pressure, density, temperature and species mass-fraction distributions along any radial line.
- Pressure coefficients, skin-friction, heat transfer rates and the species mass-fraction vaiations over body surface along a meridian line.
- Grid system.
- Constant pressure, density, temperature and species mass-fraction contours.
- Velocity vectors.

In the first card a user can punch the desired radial line numbers continuously. Each line number should be punched in a three-column width field. A blank field or a zero line number skips further informations. A blank card avoids radial distribution plotting. Any other graphs can be plotted by punching a non-zero digit in the appropriate slot in the next card. A total of 11 numbers will be read from this card, assuming that it is prepared for two-column field widths. The sequence of these numbers corresponds to distributions along the body, grid system, 8 contour maps and the plot for velocity vectors, respectively.

Example 1:

bb1bb3bb8b22b35b55b57b58b59b60 [card 1]
b1b0b1b1b1b1b1b1b1b1b1 [card 2]

Example 2:

bb1b59 [card 1]
b1b1b1b1b1b0b0b0b0b0b0b1 [card 2]

Three-Dimensional Graphs

HYREFS-GR3 can plot the following graphs:

- Pressure, density, temperature and species mass-fraction distributions along any radial line.
- Pressure coefficients, skin-friction, heat transfer rates and the species mass-fraction variations along the body surface in any meridian plane.
- Grid system on any meridian plane.
- Constant pressure, density, temperature and species mass-fraction contours on any meridian plane.
- Axial flow velocity vectors in any meridian plane.
- Cross flow velocity vectors in any ξ=constant section.

The first card should carry the meridian plane number (K value) in its first two columns. Plots on both windward and leeward sides of this plane will be drawn. In the second card a user can punch the desired radial line numbers (I values) in the same way as is used for 2-d graphs. A blank card avoids radial distribution plotting. Any other graphs except the cross flow can be plotted by punching a non-zero digit in the appropriate slot in the next card. All the 11 numbers will represent the same graphs that are mentioned for 2-d graphs. Cross flow velocity vectors can be plotted for the desired ξ=constant surfaces by punching I numbrs in the fourth card.

Example 1: To draw graphs in the plane of symmetry;

bb2 [card 1]
bb1bb3bb8b22b35b55b57b58b59b60 [card 2]
b1b0b1b1b1b1b1b1b1b1b1 [card 3]
b35b45b55b57b58b59 [card 4]

5. Sample Calculations

HYREFS codes are proved to be capable of simulating very complex flows. A number of solutions have been presented in the listed References 5, 10 and 14. Therefore, it will be unnecessary to describe those solutions once again in this manual. On the other hand, it would be much more helpful if we present some typical solutions that can be reproduced by any user even without having an understanding on the code structure. In this section we describe how to get acquainted with HYREFS codes through a series of sample calculations.

5. 1. Generating Initial Data

HYREFS-NR2 is capable of developing solutions by generating the necessary initial data by itself. These solutions can then be fed into the other codes for simulating desired flows.

Sample 1

Frozen flow around a hemisphere-cone body is generated for a freestream Mach number equal to 25. Freestream pressure and density are taken from the standard atmosphere at an altitude of 70 km. The following input cards have been prepared:

Code: HYREFS-NR2; Mesh: 46 × 30

```
1.0,1.8,20.0,5.0,16,30,54                          [card 1]
25.0,0.0,70.0,5.5205,8.7535D-05,20.0               [card 2]
2,0                                                [card 3]
−1,−1,−1,0.0                                       [card 4]
0.3,0.3,0.4                                        [card 5]
0                                                  [card 6]
3000,4,1.0D-10,0.0,0                               [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0                         [card 8]
```

Before executing HYREFS-NR2 code for these data cards, it has been edited for a 46 × 30 mesh system. On a Fujitsu VP200 supercomputer, this run took 130 seconds cpu time and the residual has dropped down to 4.237E-10 after 3000 iterations. The solution is stored on a file that is allocated to the Unit 12. Memory locations used by this job are 792 kilobites.

The stored data are read again for plotting various graphs using HYREFS-GR2 code, which are shown in Figs. 9 through 12. The input cards used for these plots are listed below:

Code: HYREFS-GR2

```
bb2bb0                                             [card 1]
b1b1b1b0b0b0b0b0b0b0b0                             [card 2]
```

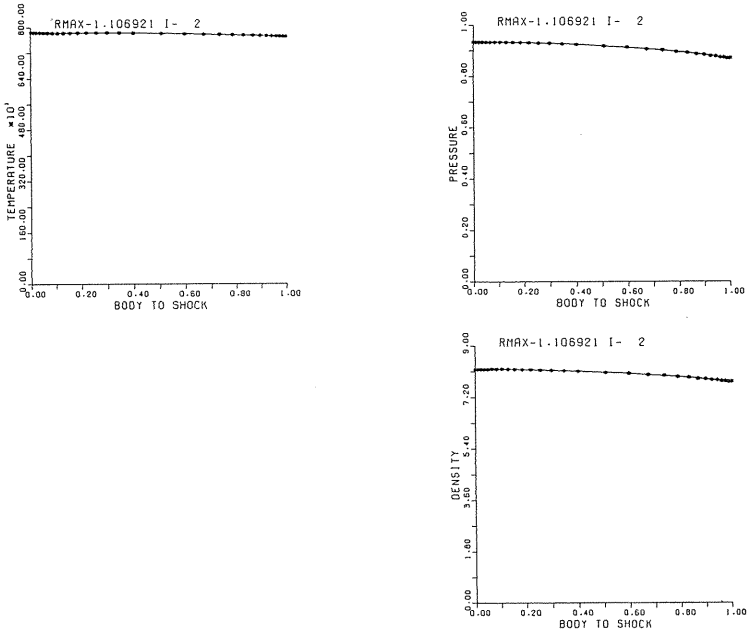

Fig. 9. Radial Plots from Sample 1 Calculation, M_∞=25.

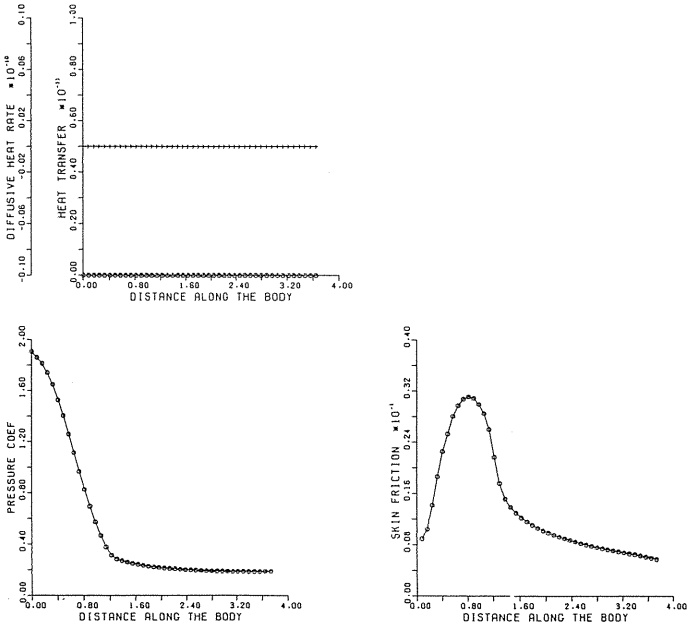

Fig. 10. Pressure Coefficient and Skin-friction along the Body, Sample 1 Calculation, M_∞=25.

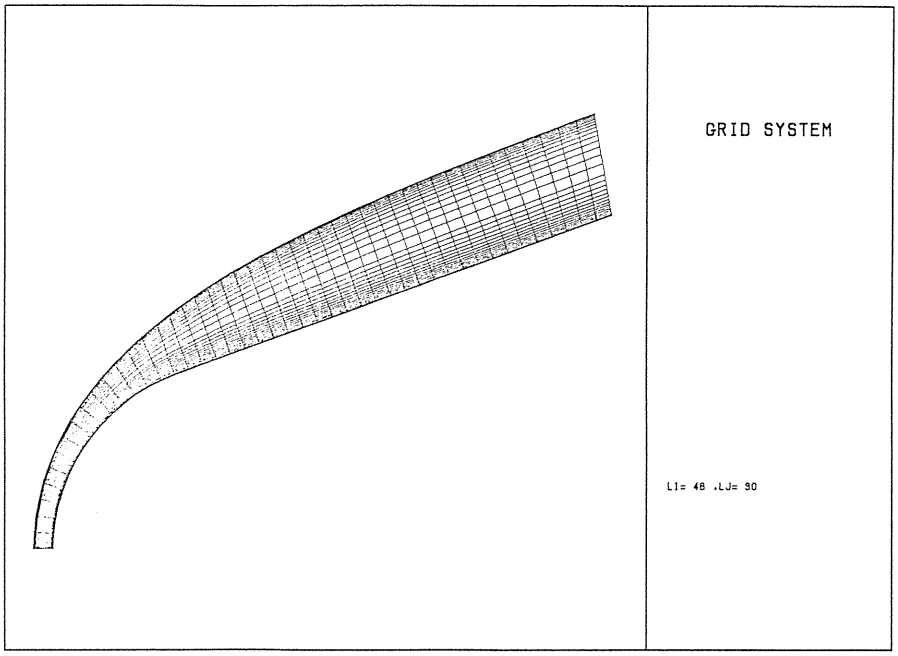

Fig. 11. Grid System from Sample 1 Calculation, M_∞=25.

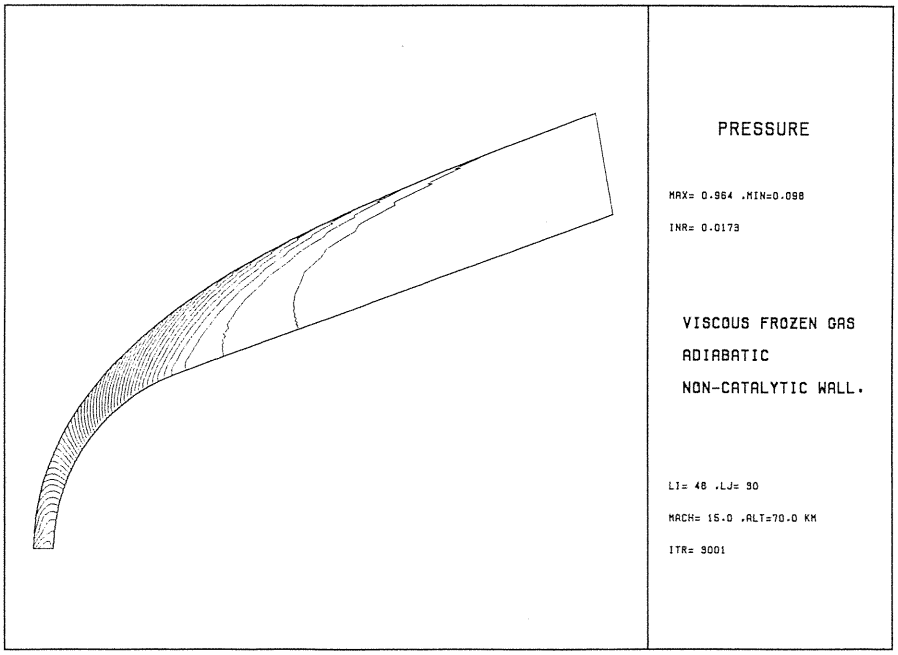

Fig. 12. Constant Pressure Contours Obtained from Sample 1 Calculation, M_∞=25.

Sample 2

A frozen flow around a hemisphere-cylinder body is obtained. Flow generating code has been edited for a 30 × 30 mesh system and the graphs drawn by the graphics code are shown in Figs. 13 and 14. The data cards prepared for this calculation are as follows:

Code: HYREFS-NR2; Mesh: 30 × 30

```
1.0,1.0,00.0,9.25,20,00,100                    [card 1]
15.0,0.0,70.0,5.5205,8.7535D-05,20.0           [card 2]
2,0                                            [card 3]
—1,—1,—1,0.0                                   [card 4]
0.3,0.3,0.4                                    [card 5]
0                                              [card 6]
3000,4,1.0D-10,0.0,0                           [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0                     [card 8]
```

Code: HYREFS-GR2

```
bb0                                            [card 1]
b1b0b0b1b0b0b0b0b0b0b0b0                       [card 2]
```

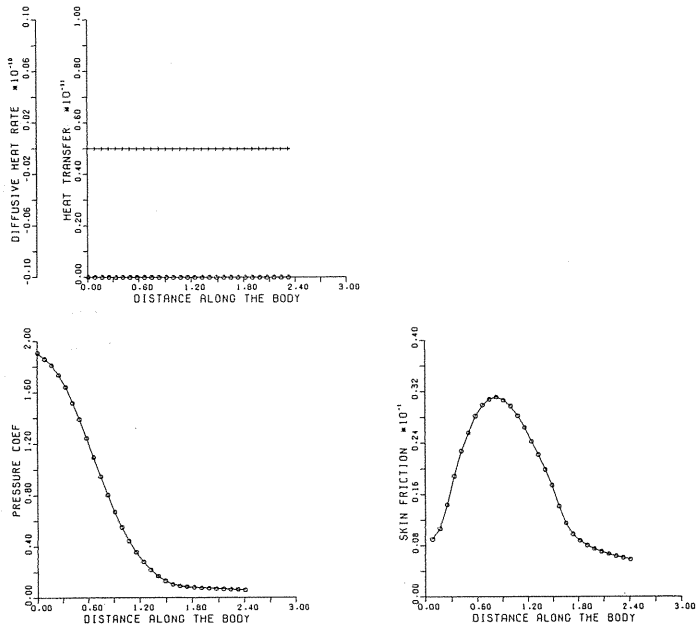

Fig. 13. Pressure Coefficient and Skin-friction along the Body, Sample 2 Calculation, M_∞=15.

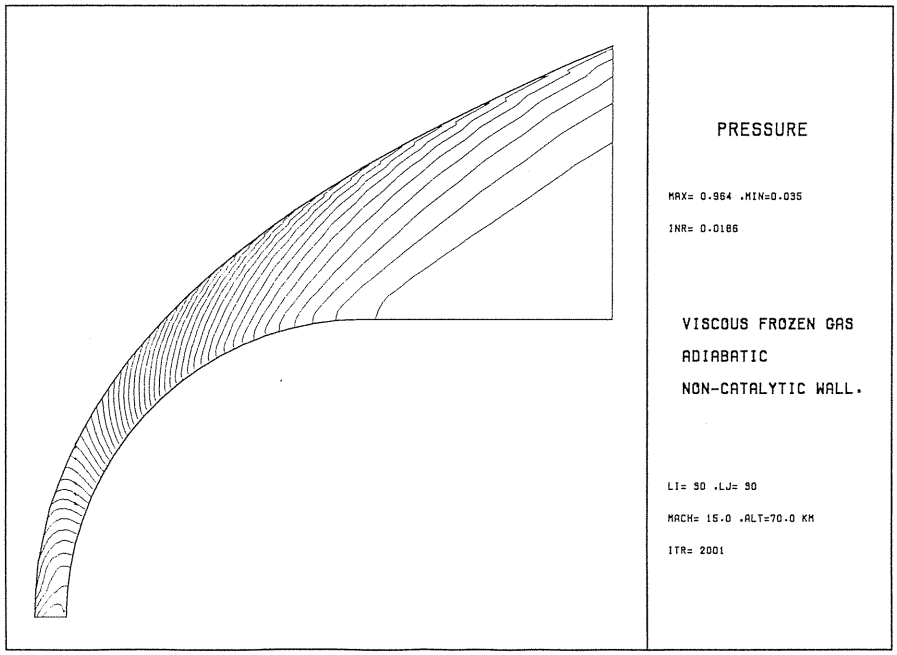

Fig. 14. Constant Pressure Contours Obtained from Sample 2 Calculation, M_∞=15.

5. 2. *Increasing the Body Length*

Often it is necessary to generate flows over a long body which is extended from a short one in a few stages rather than in just one attempt. In such a case, data over a short body can be supplied as initial data to a larger-scale calculation.

Sample 3

Solution over a 8.78 meter-long hemisphere-cylinder body is obtained by gradually extending the body length, starting from the solution generated in Sample 2. The grid size in ξ direction is increased from 30 to 100 in seven stages. Each calculation is made for 2000 iterations. The final results are shown in Figs. 15 through 17. Data cards that are used for the final run (extending a 90 × 30 body to a 100 × 30 body) are as follows:

Code: HYREFS-NR2; Mesh: 100 × 30

```
1.0,1.0,00.0,9.25,20,00,100                    [card 1]
15.0,0.0,70.0,5.5205,8.7535D-05,20.0           [card 2]
2,0                                            [card 3]
—1,—1,—1,0.0                                   [card 4]
0.3,0.3,0.4                                    [card 5]
1                                              [card 6]
2000,4,1.0D-10,0.0,0                           [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0                     [card 8]
```

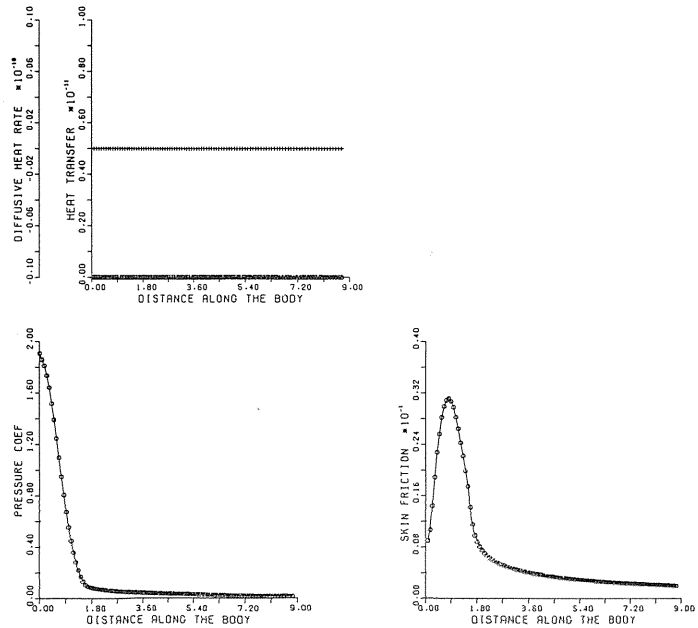

Fig. 15. Pressure Coefficient and Skin-friction along the Body, Sample 3 Calculation, M_∞=15.

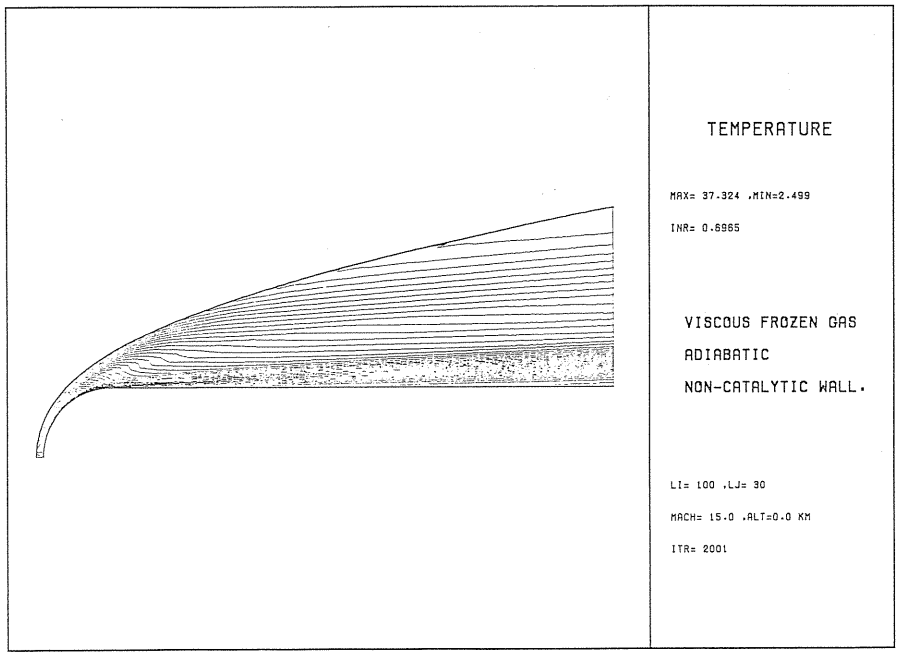

Fig. 16. Constant Temperature Contours Obtained from Sample 3 Calculation, M_∞=15

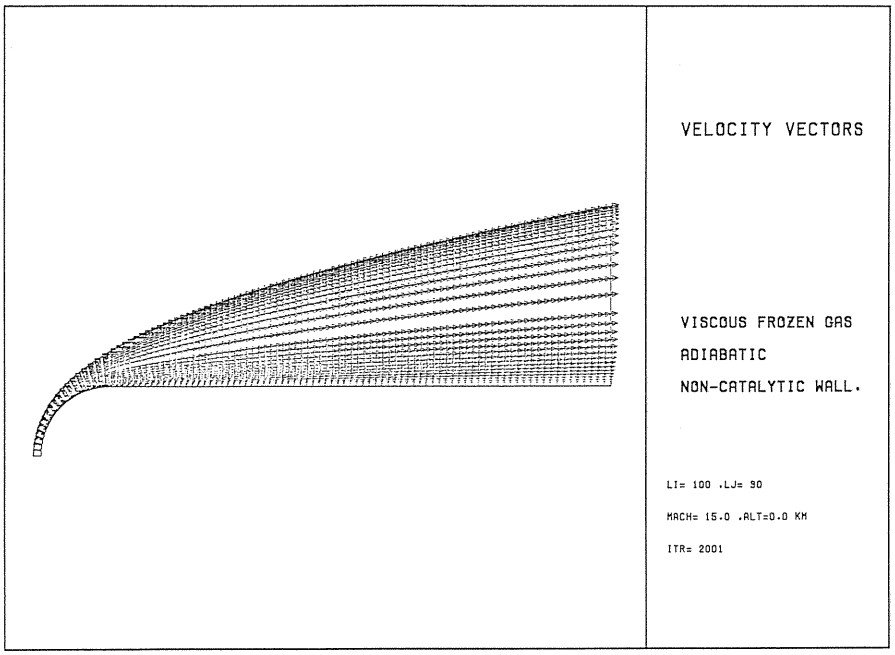

Fig. 17. Velocity Vectors Obtained from Sample 3 Calculation, M_∞=15.

HYREFS-NR2 has been edited for a 100 × 30 mesh system. This run took only 183 seconds.

Code : HYREFS-GR2

bb0 [card 1]
b1b0b0b1b1b0b0b0b0b0b0 [card 2]

5. 3. *Two-Dimensional Reacting Flows*

Once an appropriate frozen flow is obtained, reacting flows can be easily simulated using HYREFS-CR2 code. Flows over a hemisphere-cone-cylinder body with a freestream Mach number equal to 25 are generated for two different boundary conditions, using the extended initial data that were generated after Sample 1 calculation.

Sample 4

The data cards used for an adiabatic wall calculation are listed below and the graphs obtained using graphics code are shown in Figs. 18 through 21.

Code : HYREFS-NR2 ; Mesh : 48 × 30

1.0,1.8,20.0,5.0,16,30,54 [card 1]
25.0,0.0,70.0,5.5205,8.7535D-05,20.0 [card 2]

3,1 [card 3]
—1,—1,—1,0.0 [card 4]
0.3,0.3,0.4 [card 5]
1 [card 6]
2000,9,1.0D-10,0.0,0 [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0 [card 8]

The above calculation is performed on a 48 x 30 mesh system. For 2000 iterations a VP200 computer took 430 seconds and used 1716 kilobites memory.

Code : HYREFS-GR2

bb2b20 [card 1]
b1b0b0b0b1b0b0b0b0b0b0 [card 2]

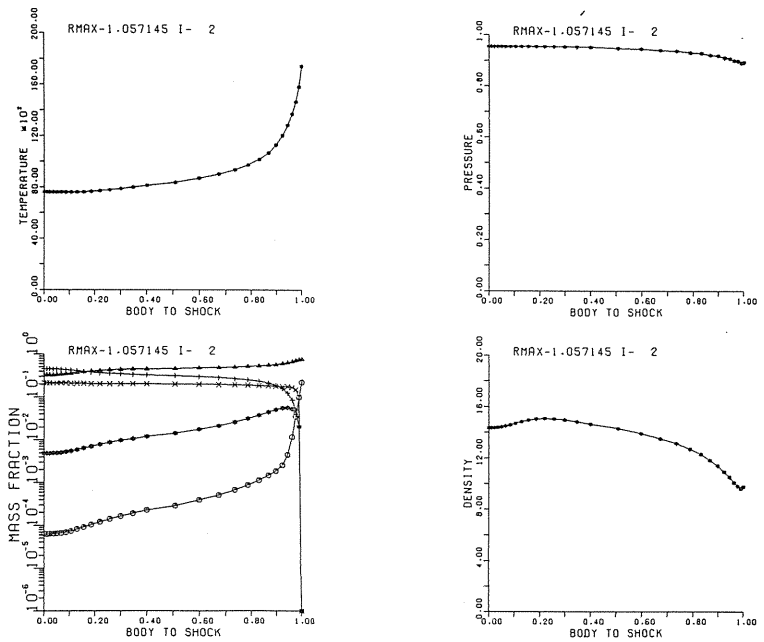

Fig. 18. Radial Plots from Sample 4 Calculation, I=2, M_∞=25 (Adiabatic Wall).

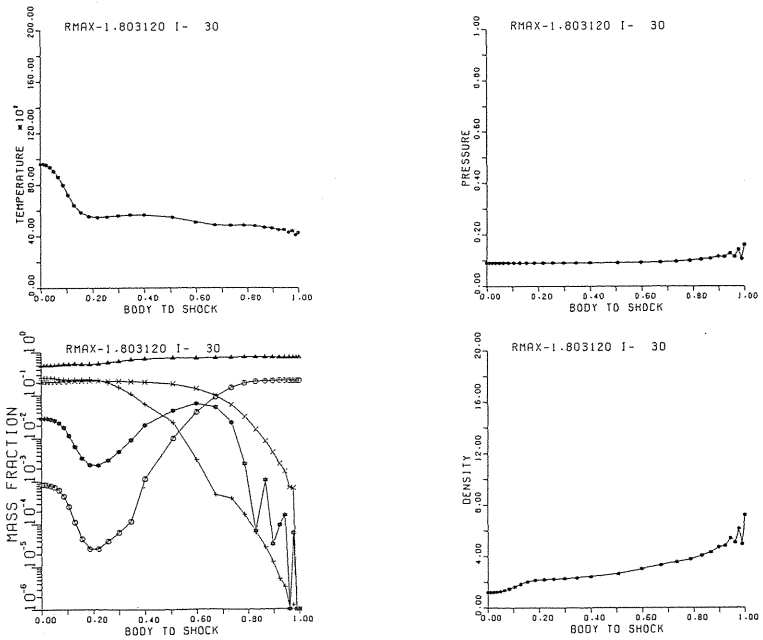

Fig. 19. Radial Plots from Sample 4 Calculation, I=30, M_∞=25 (Adiabatic Wall).

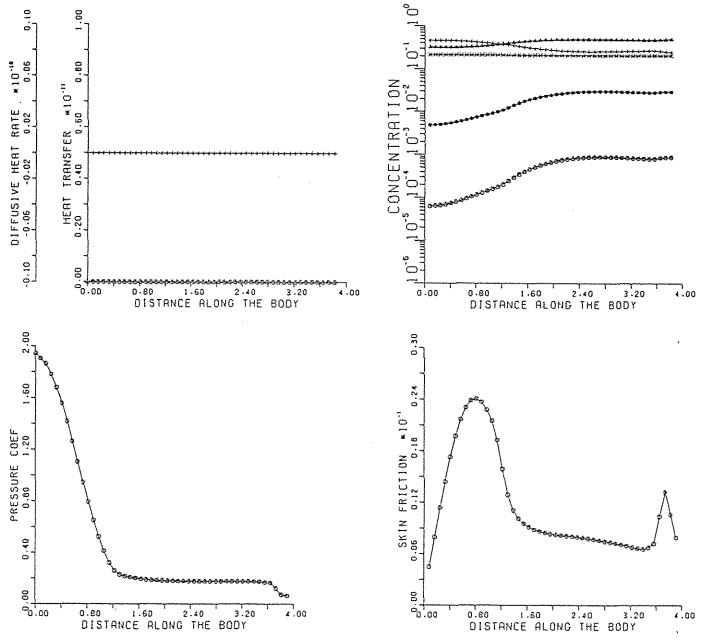

Fig. 20. Pressure Coefficient and Skin-friction along the Body, Sample 4 Calculation, Adiabatic Wall, M_∞=25.

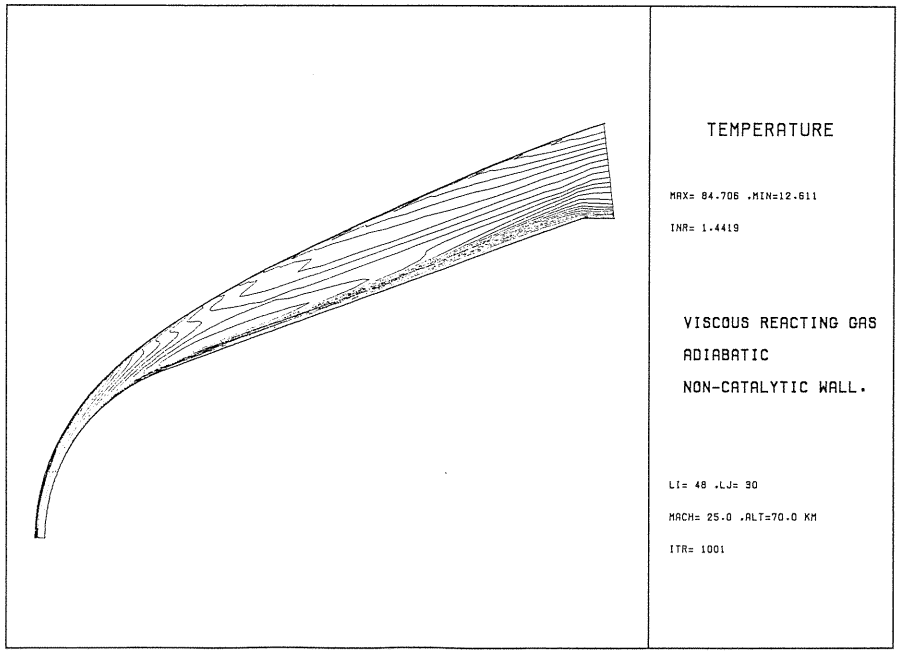

Fig. 21. Constant Temperature Contours Obtained from Sample 4 Calculation, Adiabatic Wall, M_∞=25.

Sample 5

Using the same initial data, calculations were repeated but this time for an isothermal and fully catalytic wall. The wall temperature is set to 1200 K. The data cards for this run are

Code : HYREFS-NR2 ; Mesh : 48 × 30

1.0,1.8,20.0,5.0,16,30,54	[card 1]
25.0,0.0,70.0,5.5205,8.7535D-05,20.0	[card 2]
3,1	[card 3]
−1,+1,+1,1200.0	[card 4]
0.3,0.3,0.4	[card 5]
1	[card 6]
2000,9,1.0D-10,0.0,0	[card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0	[card 8]

Solution in the form of graphs is shown in Figs. 22 through 24.

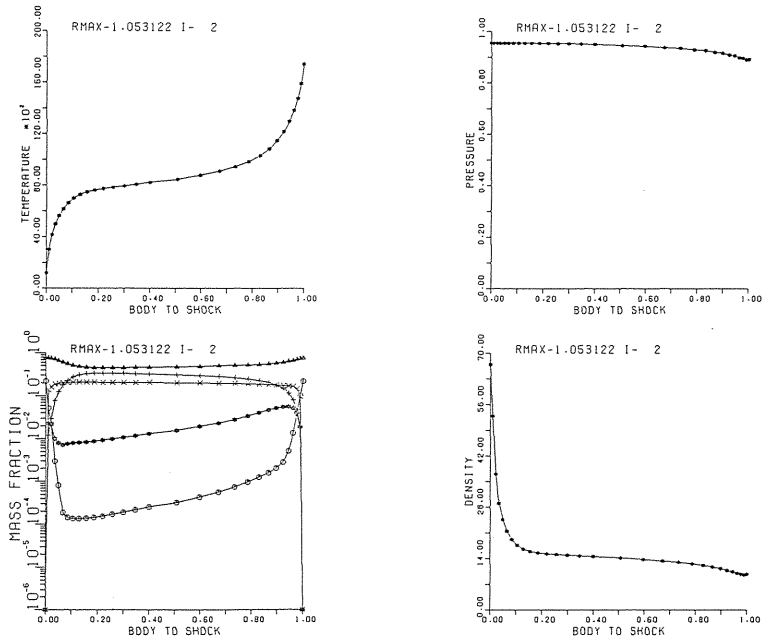

Fig. 22. Radial Plots from Sample 5 Calculation, I=2, M_∞=25 (Isothermal and Fully Catalytic Wall).

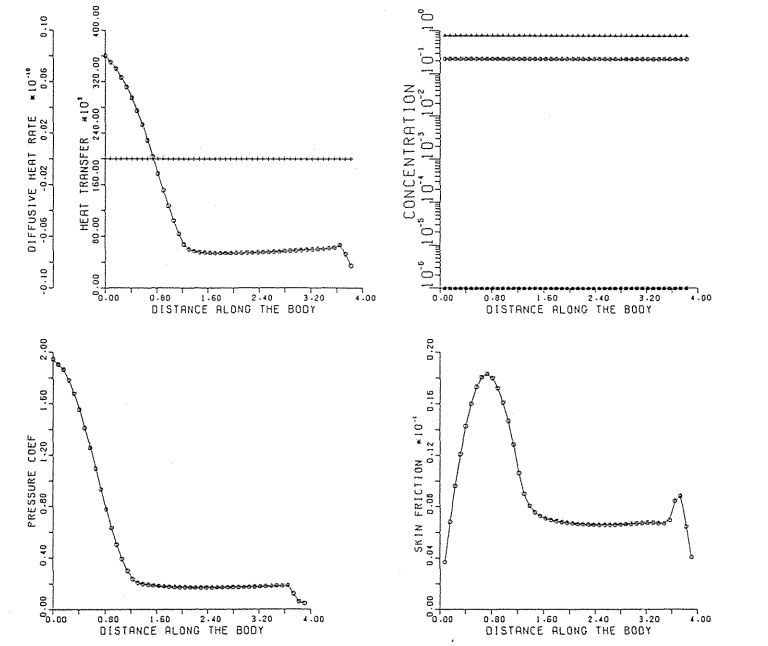

Fig. 23. Pressure Coefficient and Skin-friction along the Body, Sample 5 Calculation, Isothermal and Fully Catalytic Wall, M_∞=25.

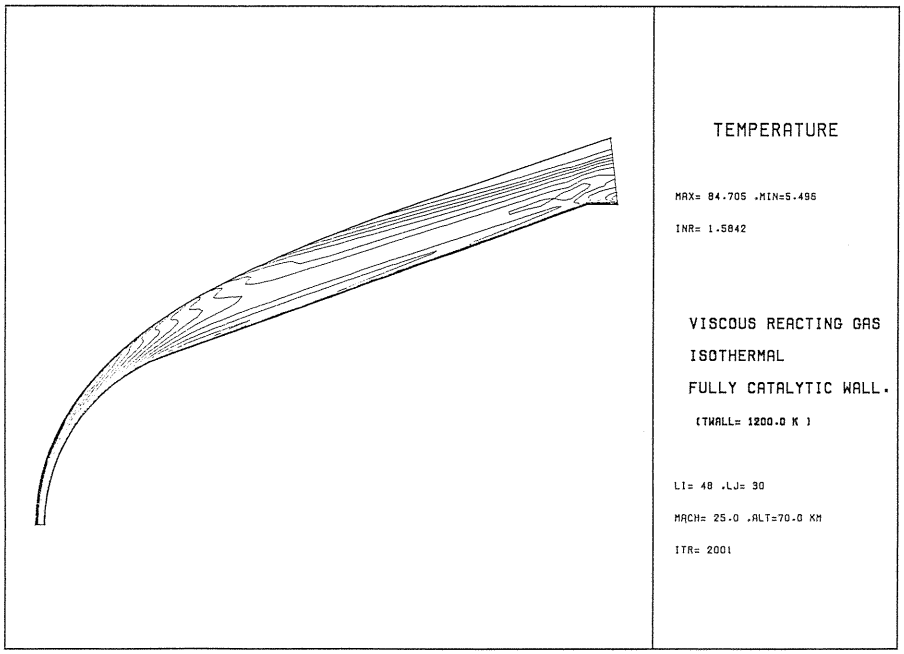

Fig. 24. Constant Temperature Contours Obtained from Sample 5 Calculation, Isothermal and Fully Catalytic Wall, M_∞=25.

5. 4. Three-Dimensional Non-reacting Flows

It won't take long time to realize the fact that simulation of a three-dimensional flow is rather difficult. Though these codes are developed to give solutions requiring minimum interactions with a user, quite often it is necessary to pay attention to grid clustering, CFL number, truncation on local time steps, choosing a compatible initial data etc. Experience on these codes indicates that it will be a proper approach if one generates flows by gradually increasing the angle of attack. However, the solutions presented in these sample calculations can be reproduced straight-away.

Sample 6

Non-reacting flow at Mach number 15 and angle of attack 5 degrees is simulated over a hemisphere-cone-cylinder body as an example. A converged axisymmetric solution generated by HYREFS-NR2 is used as initial data. The input cards are as follows:

Code: HYREFS-NR3; Mesh: 50 × 30 × 20

```
1.0,1.8,20.0,5.0,16,30,54                          [card 1]
15.0,5.0,70.0,5.5205,8.7535D-05,20.0               [card 2]
12,0                                               [card 3]
```

-1,-1,-1,0.0 [card 4]
0.3,0.3,0.4 [card 5]
2 [card 6]
1000,20,1.0D-10,0.0,1 [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0 [card 8]

This run took 16.2 minutes cpu time and used 20 megabites memory locations. The graphs are then plotted from the stored data using HYREFS-GR3 Code and are shown in Figs. 25 through 27. The following three data cards were used :

Code : HYREFS-GR3

bb2 [card 1]
bb0 [card 2]
b1b0b1b0b0b0b0b0b0b0b0 [card 3]

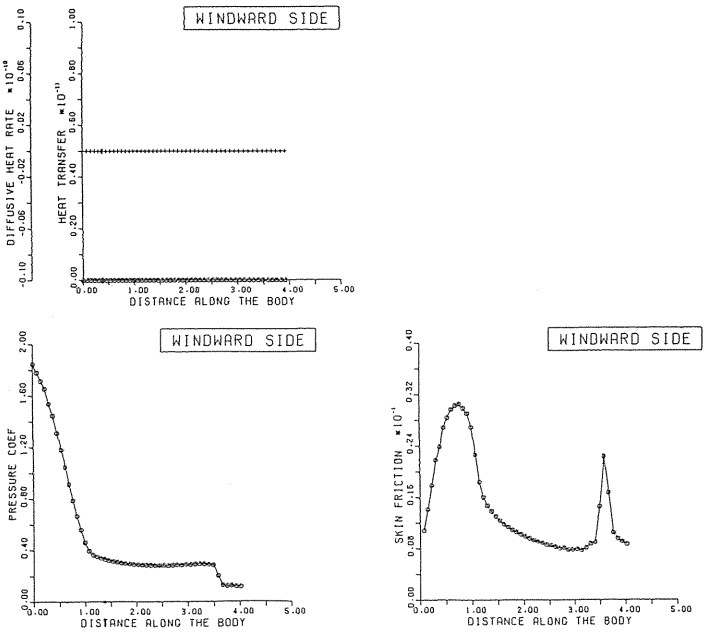

Fig. 25. Pressure Coefficient and Skin-friction along the Body in the Windward Side of Plane of Symmetry, Sample 6, M_∞=15.

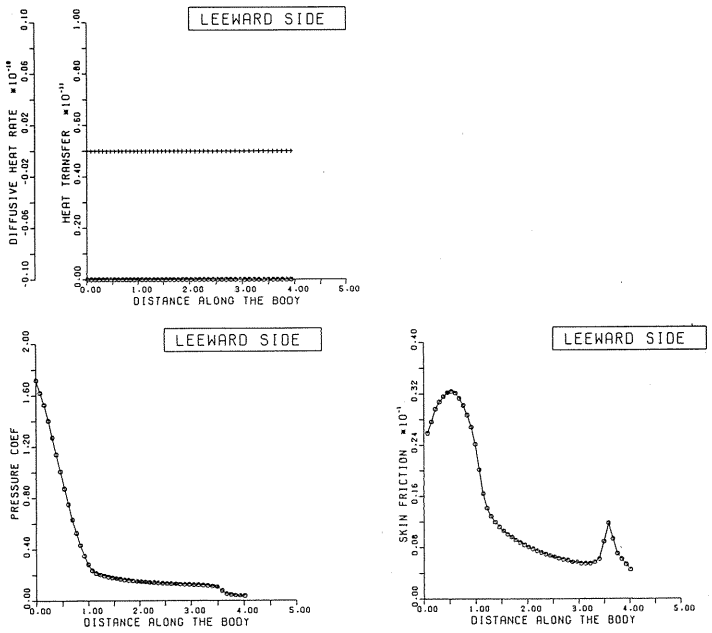

Fig. 26. Pressure Coefficient and Skin-friction along the Body in the Leeward Side of Plane of Symmetry, Smaple 6, M_∞=15.

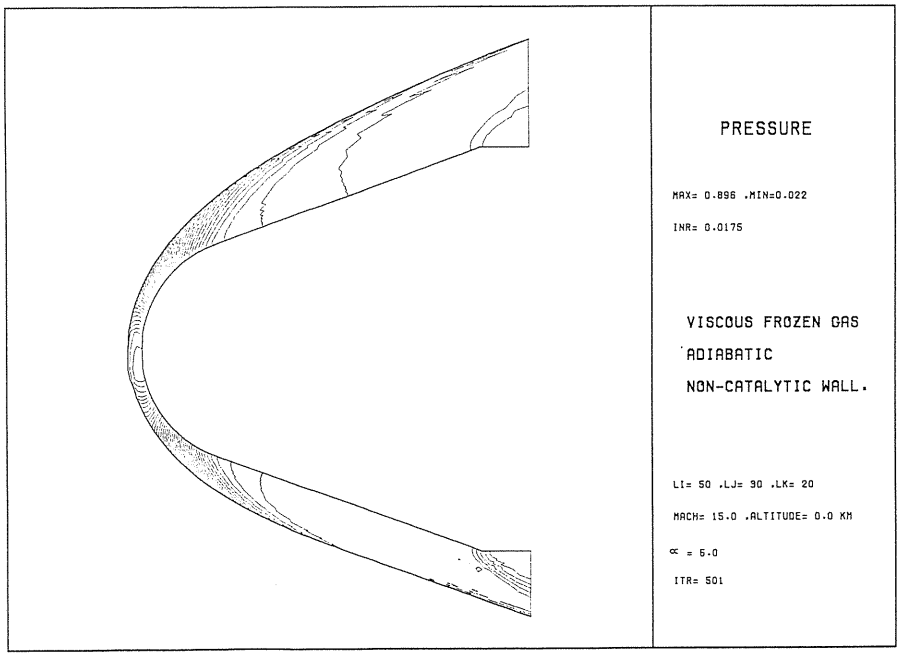

Fig. 27. Constant Pressure Contours in the Plane of Symmetry Obtained from Sample 6 Calculation, M_∞=25.

5. 5. *Three-Dimensional Reacting Flows*

Reacting flows can be simulated also by simultaneously introducing angle of attack and chemical reactions.

Sample 7

As a final example, a chemically reacting three-dimensional flow around a hemi-sphere-cylinder body is generated for an angle of attack equal to 7 degrees. This calculation is also done by feeding an axisymmetric non-reacting solution as initial data (generated in Sample 3). A 50 × 30 × 20 grid system has been used and the solution is stored after 2000 time steps. Input data cards prepared for this calculation are listed below:

Code : HYREFS-CR3 ; Mesh : 50 × 30 × 20

```
1.0,1.0,00.0,9.25,20,00,100                    [card 1]
15.0,7.0,70.0,5.5205,8.7535D-05,20.0           [card 2]
13,1                                           [card 3]
—1,—1,—1,0.0                                   [card 4]
0.3,0.3,0.4                                    [card 5]
2                                              [card 6]
2000,100,1.0D-10,0.0,1                         [card 7]
1,10,1,1,1,0,0,0,0,0,0,0,0                     [card 8]
```

This run utilized 50.164 megabites memory locations and took 104.52 minutes cpu time on a VP200 computer. The graphs plotted using HYREFS-GR3 Code are shown in Figs. 28 through 32.

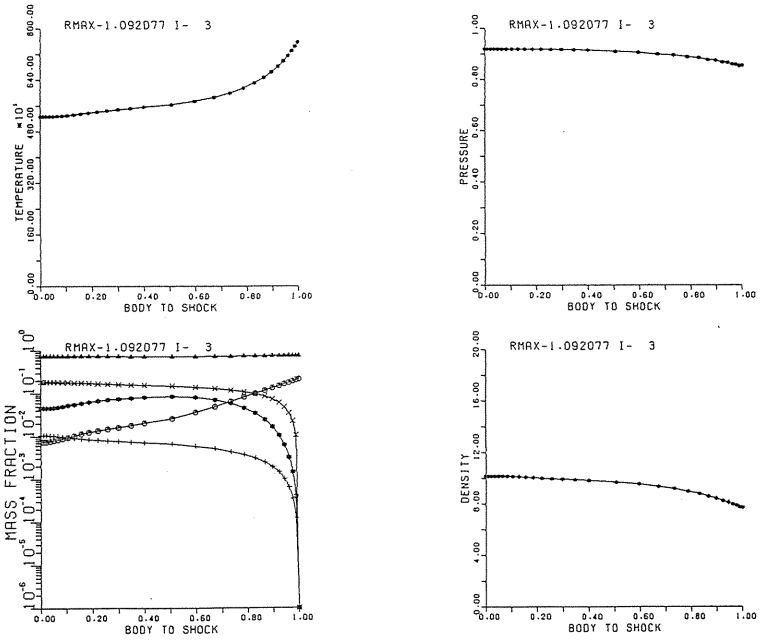

Fig. 28. Radial Plots for I=3 in the Windward Side of Plane of Symmetry, Sample 7, M_∞=15.

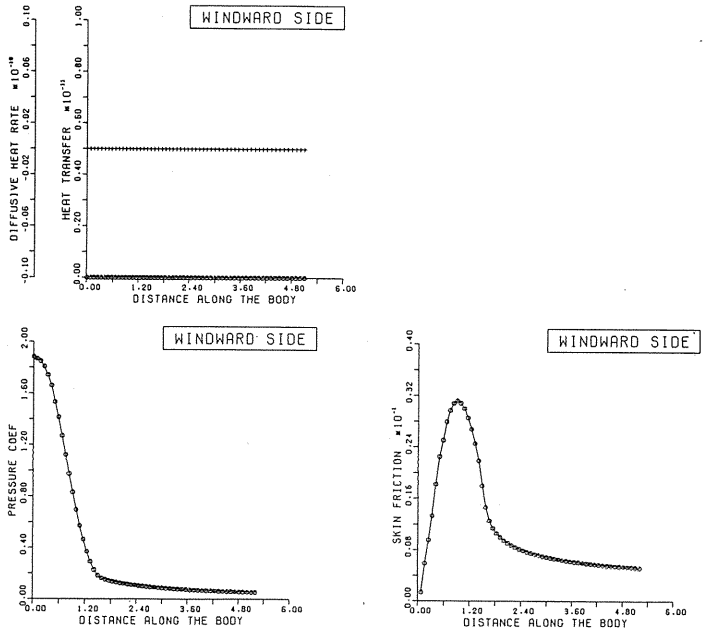

Fig. 29. Pressure Coefficient and Skin-friction along the Body in the Windward Side of Plane of Symmetry, Sample 7, M_∞=15.

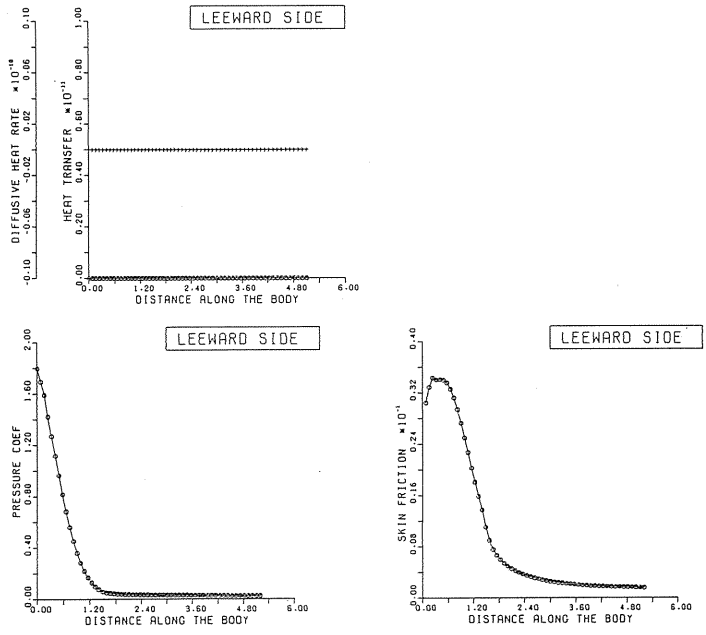

Fig. 30. Pressure Coefficient and Skin-friction along the Body in the Leeward Side of Plane of Symmetry, Sample 7, M_∞=15.

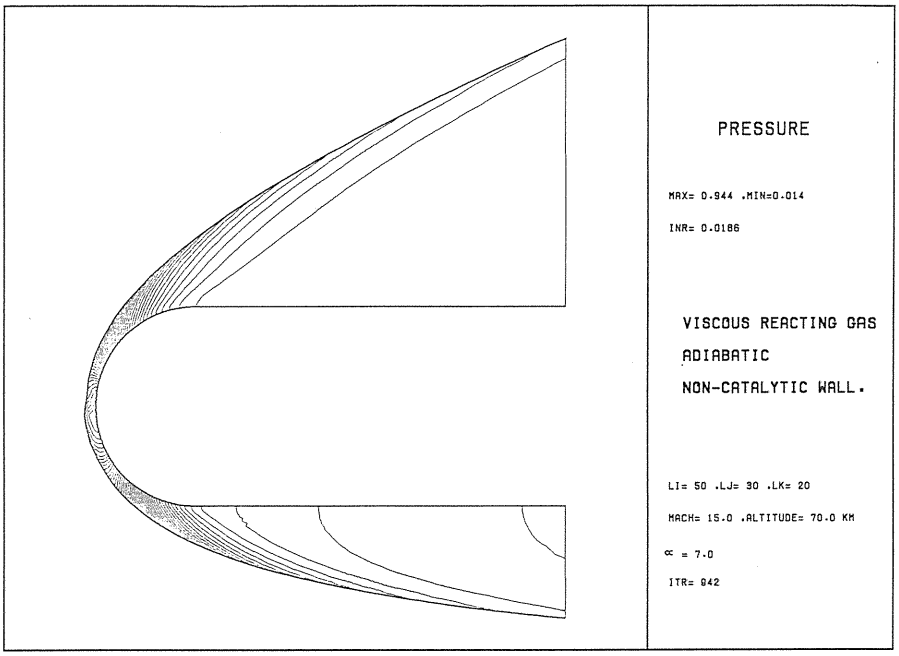

Fig. 31. Constant Pressure Contours in the Plane of Symmetry Obtained from Sample 7 Calculation, M_∞=25.

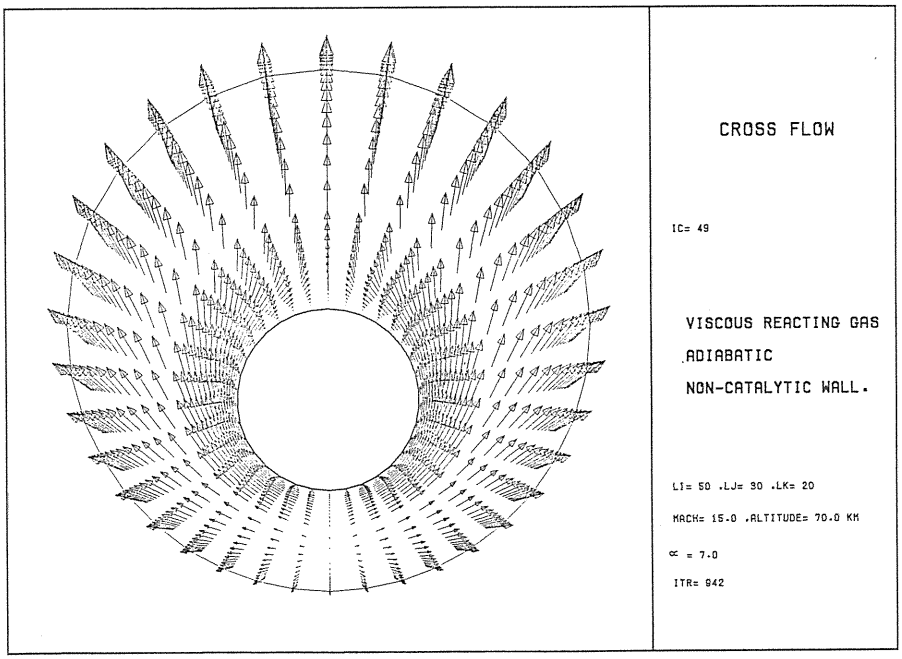

Fig. 32. Cross Flow Velocity Vectors Obtained from Sample 7 Calculation, M_∞=25.

6. Acknowledgements

The research work described in this Manual is an outcome of the two years' efforts between December 1986 and March 1989 and it has been carried out under the auspice of several sponsored projects. The authors are grateful to the Computation Center of Nagoya University for encouraging this work and for donating a few tens of hours of computer time under the "Code Development Program". The authors are also grateful to Dr. T. Ishiguro and Dr. T. Ogawa of Numerical Simulation Center, National Aerospace Laboratory, for allowing us to use their Fujitsu VP400 supercomputer. Finally but never to the least, the first author wishes to express his heartiest thanks to the Management of National Aerospace Laboratory, Tokyo, for providing him a stimulating environment during his entire stay over there.

References

1) Park, C., On Convergence of Computation of Chemically Reacting Flows, AIAA Paper No. 85-0247, January 85.
2) Candler, G., Park. C., The Calculation of Radiation from Nonequilibrium Hypersonic Flows, AIAA Paper No. 88-2678, June 88.
3) Lee, J. H., Basic Governing Equations for the Flight Regimes of Aeroassisted Orbital Transfer Vehicles, Thermal Design of Aeroassisted Orbit Transfer Vehicles, ed. H. F. Nelson, Progress in Aeronautics and Astronautics, Vol. 96, 1985, pp. 5–53.
4) Prabu, D. K., Tannehill, J. C. and Marvin, J. G., A New PNS Code for Three-Dimensional Chemically Reacting Flows, AIAA Paper No. 87-1472, June 87.
5) Reddy, K. V., Fujiwara, T., Ogawa, T. and Arashi, K., Computation of Three-Dimensional Chemically Reacting Viscous Flow around the Rocket Body, AIAA Paper No. 88-2616, June 1988. Also in Institute of Space and Astronautical Science (ISAS) Report SP-7, 1988, pp. 39-71.
6) Kuo, K. K., Princeples of Combustion, John Wiley & Sons, Sons, 1986, pp. 161–227.
7) Moss, J. N., Reacting Viscous-Shock-Layer Solutions with Multicomponent Diffusion and Mass Injection, NASA TR-411, June 1974.
8) White, F. M., Viscous Fluid Flow, McGraw-Hill Inc., New York, 1974, pp. 28–36.
9) Beam, R. and Warming, R. F., An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, AIAA J, Vol. 16, No. 4, April 1978.
10) Reddy, K. V. and Fujiwara, T., Computation of Hypervelocity Three-Dimensional Reacting Flows, Memoirs of the Faculty of Engineering, Nagoya University, Vol. 39, No. 2, November 1987, pp. 396–412.
11) Pulliam, T. H. and Steger, J. L., Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow, AIAA J., Vol. 18, No. 2, February 1980.
12) Vincenti, W. G. and Kruger, Jr. C. H., Introduction to Physical Gas Dynamics, John Wiley & Sons Inc., 1965, pp. 265–273.
13) Rizzi, A. W. and Inouye, M., Time-Split Finite-Volume Method for Three-Dimensional Blunt-Body Flow, AIAA J., No. 11, November 1973.
14) Reddy, K. V. and Fujiwara, T., A Hybrid Technique for Hypervelocity Flows around a Reentry Vehicle with Real Gas Effects, Memoirs of the Faculty of Engineering, Nagoya University, Vol. 39, No. 2, 1987, pp. 306–323.