Final Report for:

Modeling of Heavy-Gas Effects on Airfoil Flows

548

Submitted to: NASA Ames Research Center

Technical Monitor: Raymond Hicks
Computational Aerodynamics Group

NASA Grant: NAG2-708

Submitted by: Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139

Principal Investigator: Mark Drela
Associate Professor
MIT Aeronautics and Astronautics

11 May 92

1 Summary

A non-ideal gas model has been developed and retro-fitted into the MSES viscous/inviscid multi-element airfoil program. The specific applications targeted are compressible airfoil flows in wind tunnels employing heavy gases. The particular gas modeled in this work has been sulfur hexafluoride (SF_6), although most heavy gases could be implemented if adequate state and caloric data were available.

Numerical predictions with MSES indicate that the non-ideality of SF_6 significantly influences airfoil behavior in transonic flows, especially at the higher total pressures envisioned for pressurized tunnels. The dominant effect is that for a given freestream Mach number, local Mach numbers in supersonic zones are lower, and shocks are correspondingly weakened. Another (but apparently smaller) effect is that for a given edge Mach number, a boundary layer in a heavy gas is theoretically somewhat more resistant to an adverse pressure gradient due to reduced adiabatic heating near the wall.

As pointed out by Wagner and Schmidt [1], transonic small-disturbance theory is valid for non-ideal gases. Similarity between two flows can be obtained if the transonic similarity parameter

$$K \equiv \frac{1 - M^2}{[M^2(\gamma'+1)]^{2/3}}$$

is matched, and if the pressure coefficients are scaled by the factor

$$A \equiv (\gamma'+1)\frac{M^2}{1-M^2}$$

so that the quantity AC_L must also be matched between the two flows. The parameters K and A above are defined in terms of an "equivalent" ratio of specific heats γ', which is derived in Appendix B for the second-order small-disturbance formulation employed in MSES.

Although similarity between ideal and non-ideal inviscid transonic flows is rigorous in the context of transonic small-disturbance theory, a similarity rule cannot be formulated for viscous transonic flows. In addition to the Reynolds number Re, Appendix C shows that an additional parameter γ_v is introduced. This depends on the gas properties and local Mach number, and scales the effect of the local Mach number on the displacement thickness. It therefore affects viscous displacement effects and boundary layer response to pressure gradients in compressible flows. It is highly unlikely that the parameters M, γ', Re, and γ_v can all be combined into one similarity rule for viscous transonic flows. Fortunately, numerical experiments indicate that matching K, AC_L, and Re (or M^*, C_L, and Re) still gives good correspondence between air and heavy-gas flows. Apparently, the effect of γ_v is not nearly as significant as the other three parameters.

Figure 1 compares C_p vs x/c curves for the RAE 2822 airfoil [2] at $M = 0.735$ for air, for SF_6 at 1 atm, and for SF_6 at 3 atm. Figure 2 makes the comparison at a fixed $M^* = 0.765$ instead of a fixed M. Figure 3 in turn makes the comparison at a fixed K and AC_L (corresponding to $M = 0.735$ and $C_L = 0.743$ for air). Clearly, matching M^* or K is more appropriate for evaluating

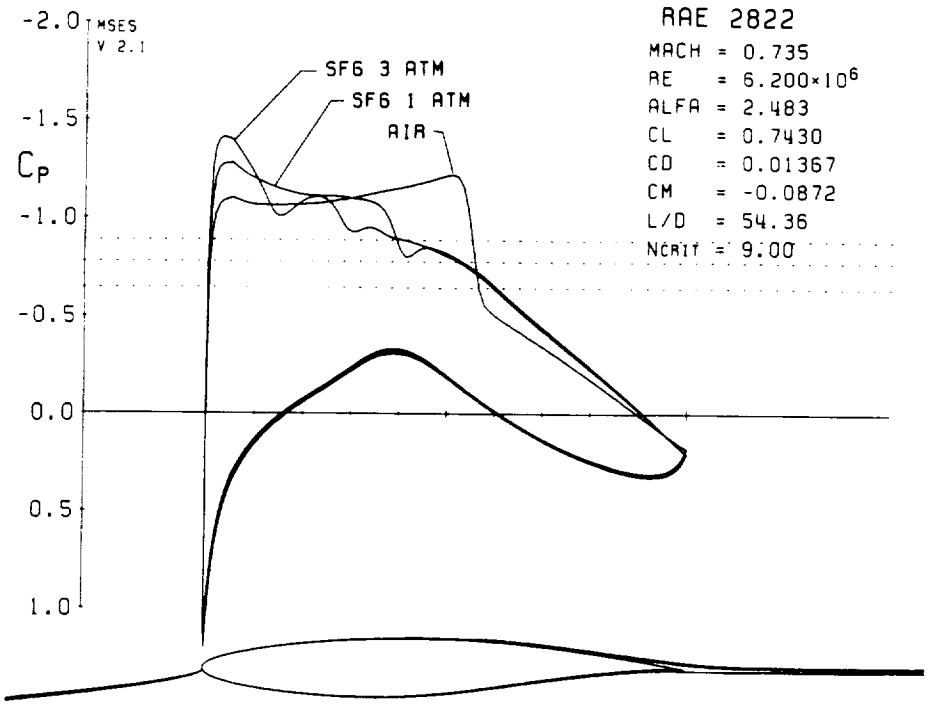

Figure 1: C_p distributions for RAE 2822 airfoil at $M = 0.735$ for air, SF_6 (1 atm), and SF_6 (3 atm). $C_L = 0.743$, $Re = 6.2$ million.

transonic flow characteristics. To illustrate further, drag-divergence behavior for air, SF_6 (1 atm), and SF_6 (3 atm) is shown versus M and M^* in Figures 4 and 5. As expected from the C_p comparisons, the effects of the type of gas on transonic drag rise are much smaller if M^* is used as the compressibility parameter in lieu of M. The Mach sweep results were not performed at fixed K and AC_L, since it is not clear how to scale the profile drag coefficient C_D over this sweep. In principle, the pressure drag should be scaled by A, while the friction drag should perhaps be left unscaled. However, it is impossible to separate these drag components in an experiment, since only the total drag is obtained from a wake survey.

For high-lift configurations, small-disturbance theory is obviously invalid, but numerical studies indicate that matching K and AC_L (or alternatively matching M^* and C_L) still gives a reasonably good match between air and heavy-gas flows. Figure 6 shows the inviscid C_p distributions over a slatted two-element airfoil described in reference [3]. A freestream Mach number of $M = 0.30$ in air produces a fairly strong shock on the slat and a somewhat weaker shock on the main element. Figure 7 compares the C_p distributions on the slat for the three gas cases at a fixed sonic Mach number $M^* = 0.3257$ (corresponding to $M = 0.30$ for air) and $C_L = 2.85$. The comparison is quite reasonable. It should be stressed again that simply matching the usual freestream Mach number $M = V_\infty/a_\infty$ and unscaled C_L gives a very poor match in all cases, except of course in effectively incompressible flows where any gas non-ideality is irrelevant.

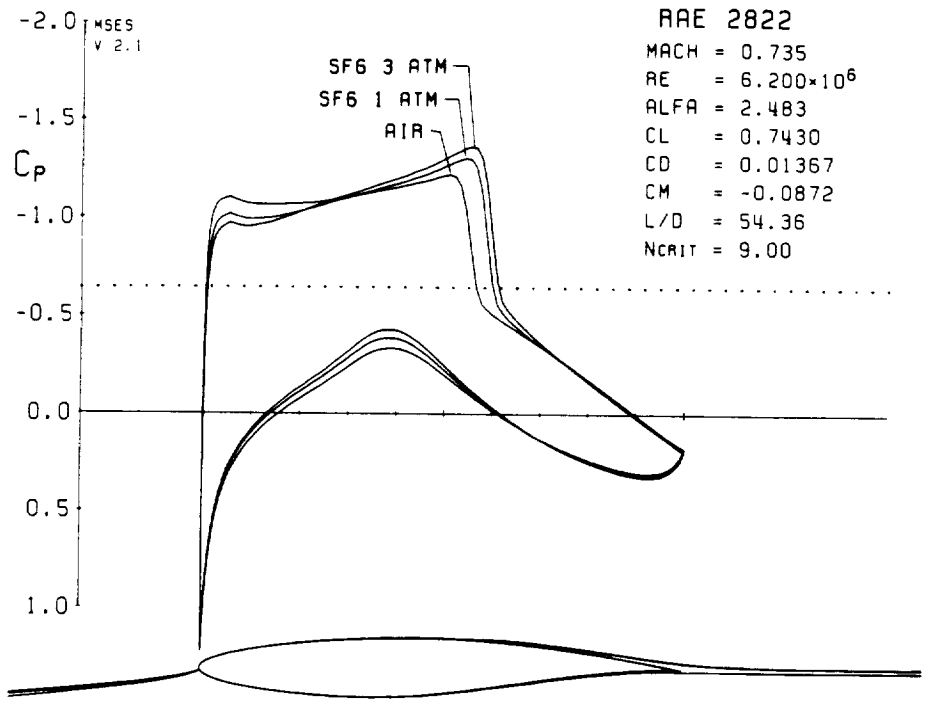

Figure 2: C_p distributions for RAE 2822 airfoil at $M^* = 0.765$ for air, SF_6 (1 atm), and SF_6 (3 atm). $C_L = 0.743$, $Re = 6.2$ million.

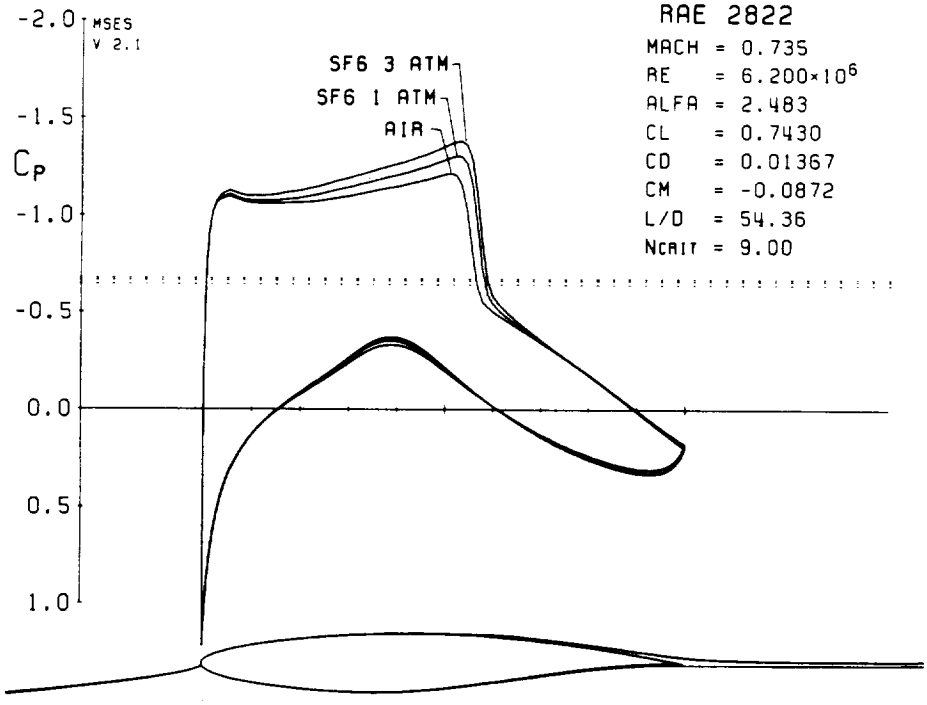

Figure 3: C_p distributions for RAE 2822 airfoil at $K = 0.3867$ for air, SF_6 (1 atm), and SF_6 (3 atm). $AC_L = 2.095$, $Re = 6.2$ million.

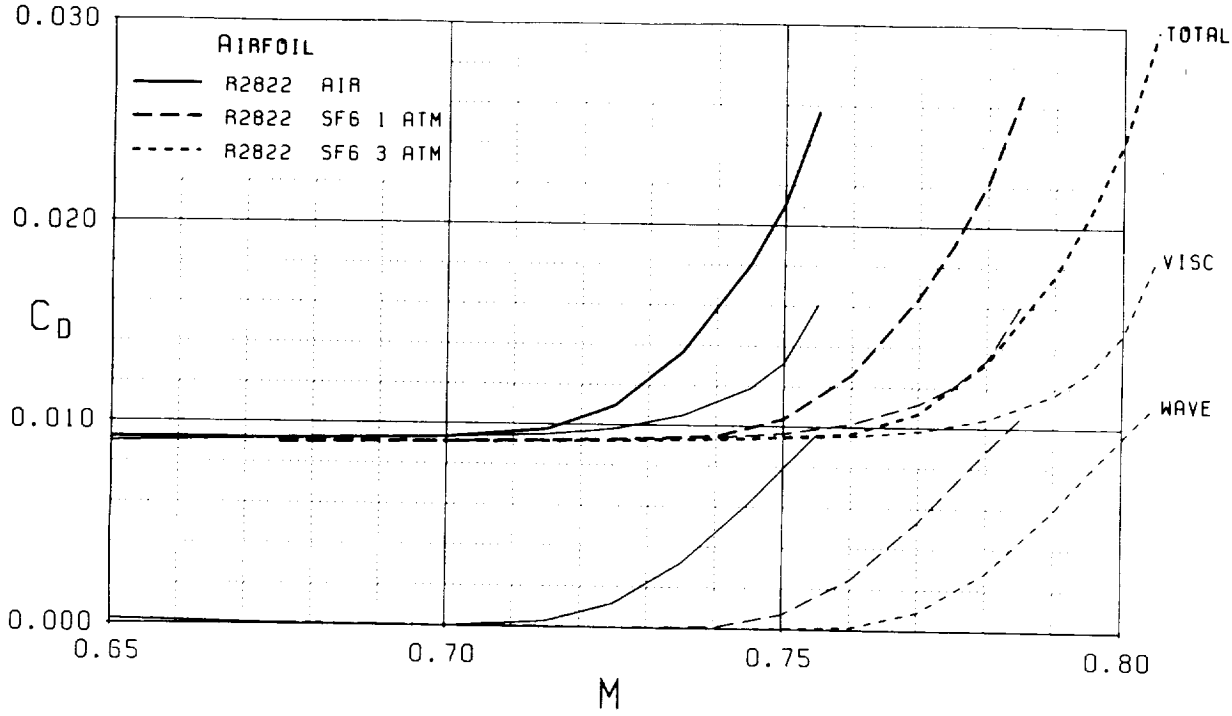

Figure 4: RAE 2822 drag-divergence behavior versus M for air, SF_6 (1 atm), and SF_6 (3 atm). $C_L = 0.743$, $Re = 6.2$ million.

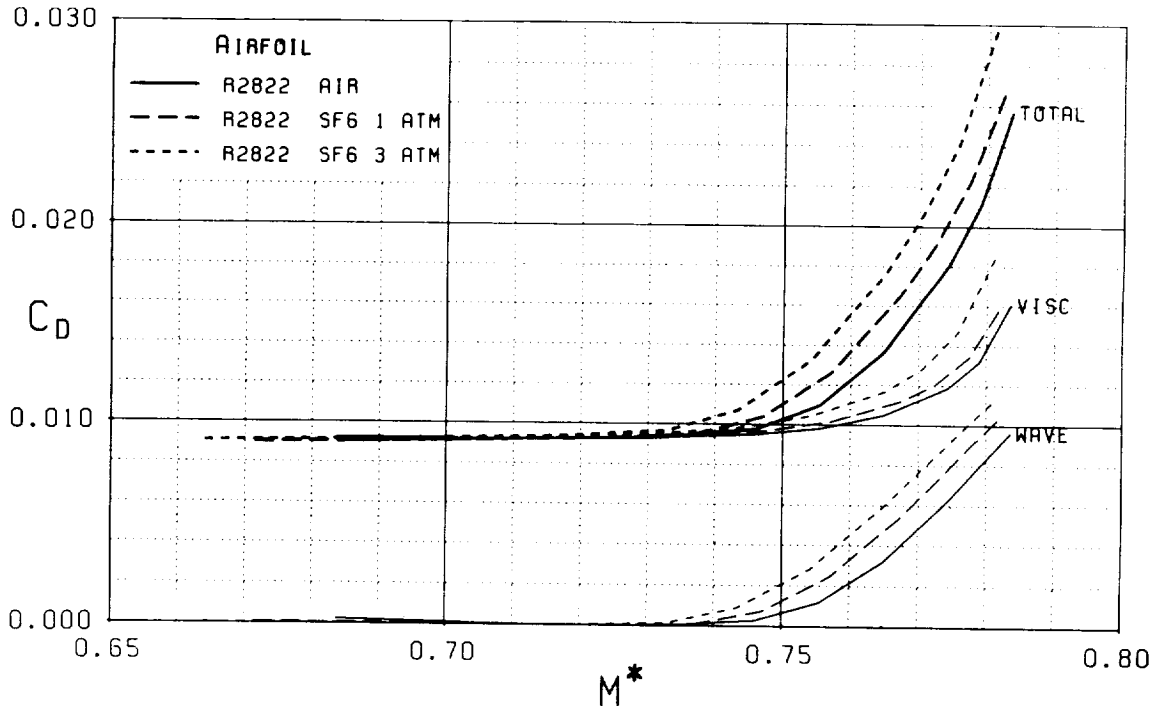

Figure 5: RAE 2822 drag-divergence behavior versus M^* for air, SF_6 (1 atm), and SF_6 (3 atm). $C_L = 0.743$, $Re = 6.2$ million.

5

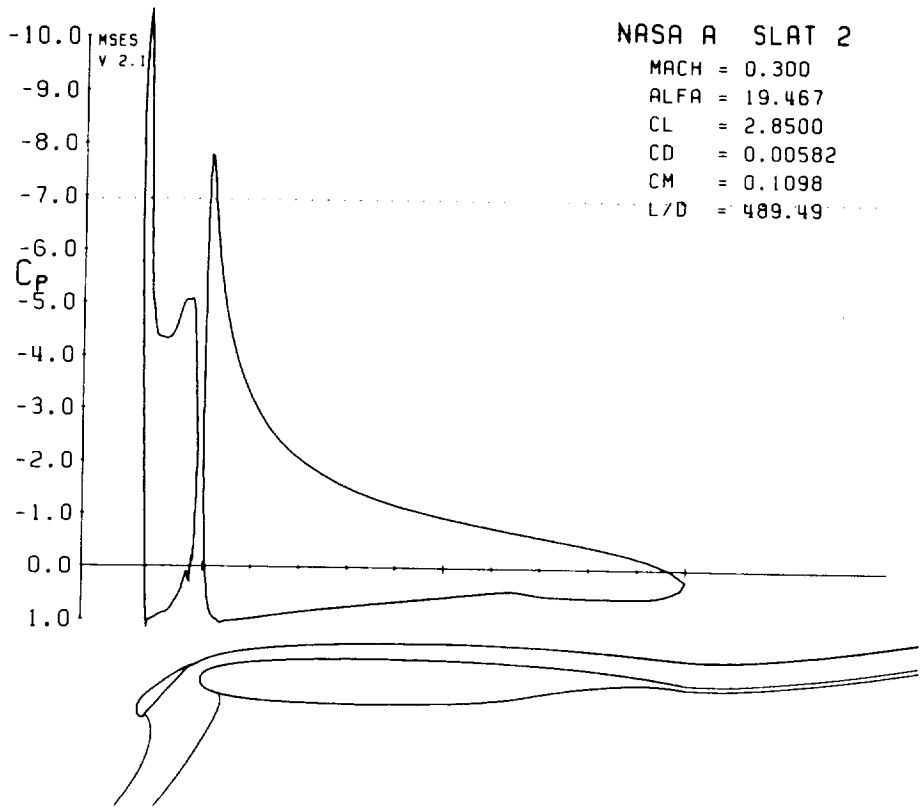

Figure 6: C_p distributions for slatted airfoil in air.

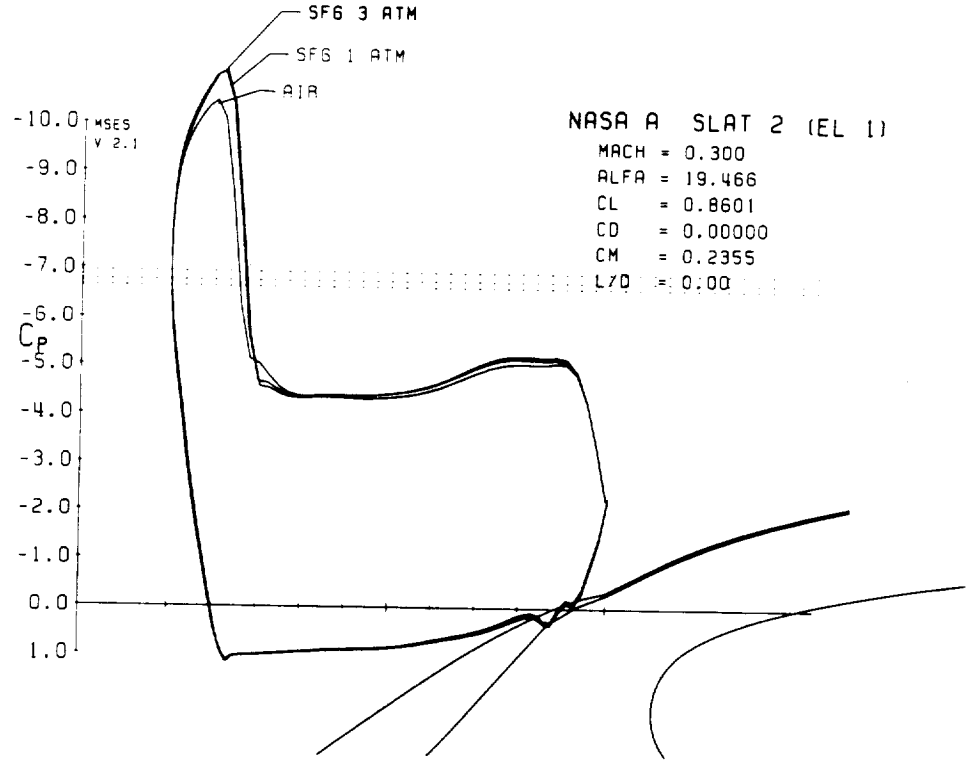

Figure 7: C_p distributions over slat at $M^* = 0.3257$ and $C_L = 2.85$ for air, SF_6 (1 atm), and SF_6 (3 atm).

The bulk of the heavy-gas model development and application to transonic, inviscid flows is documented in the SM Thesis of Marc Schafer, which is attached as Appendix A. As mentioned previously, Appendix B derives the farfield behavior of a non-ideal airfoil flow. This was required for implementation of new outer boundary conditions for the MSES code. Appendix C derives the shape parameter compressibility correction for an adiabatic boundary layer in non-ideal flow. This was required to implement new heavy-gas correlations for the MSES integral boundary layer formulation.

References

[1] B. Wagner and W. Schmidt. Theoretical investigations of real gas effects in cryogenic wind tunnels. *AIAA Journal*, 16(6), Jun 1978.

[2] P. H. Cook, M. A. McDonald, and M. C. P. Firmin. Aerofoil RAE 2822 pressure distributions and boundary layer and wake measurements. In *Experimental Data Base for Computer Program Assessment, AR-138*. AGARD, 1979.

[3] E. Omar, T. Zierten, and A. Mahal. Two-dimensional wind tunnel tests of a NASA supercritical airfoil with various high-lift systems. Contractor Report 2214, NASA, Apr 1973.

[4] M. Drela. *Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations*. PhD thesis, MIT, Dec 1985. Also, MIT Gas Turbine & Plasma Dynamics Laboratory Report No. 187, Feb 1986.

[5] J.D. Cole and L.P. Cook. *Transonic Aerodynamics*, volume 30 of *New-Holland Series in Applied Mathematics and Mechanics*. New-Holland, Amsterdam, New York, 1986.

[6] M. B. Giles and M. Drela. Two-dimensional transonic aerodynamic design method. *AIAA Journal*, 25(9), Sep 1987.

[7] D. L. Whitfield. Analytical description of the complete turbulent boundary layer velocity profile. AIAA-78-1158, 1978.

Appendix A

Modeling of Heavy Gas Effects on Airfoil Flows

by

Marc Alan Schafer

Submitted to the Department of Aeronautics and Astronautics

on May 3, 1992

in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

Thermodynamic models were constructed for a calorically imperfect gas and for a non-ideal gas. These were incorporated into a quasi one dimensional flow solver to develop an understanding of the differences in flow behavior between the new models and the perfect gas model. The models were also incorporated into a two dimensional flow solver to investigate their effects on transonic airfoil flows. Specifically, the calculations simulated airfoil testing in a proposed high Reynolds number heavy-gas test facility. The results indicated that the non-idealities caused significant differences in the flow field, but that matching of an appropriate non-dimensional parameter led to flows similar to those in air.

Thesis Supervisor: **Mark Drela,**

Associate Professor of Aeronautics and Astronautics

Acknowledgments

I would like to express my thanks to all those who made this thesis possible. First, to Mark Drela whose brilliance and ingenuity have served as an inspiration in all of my studies. Also, to Harold 'Guppy' Youngren whose leadership during the Daedalus project helped me to realize what it really means to be an engineer.

I would also like to thank my parents and the rest of my family. Without your support, I never would have made it as far as I have.

My appreciation also goes to the NASA Ames research center and the NDSEG fellowship program without whose financial support this thesis would never have happened.

4 blank

10

Contents

Abstract 3

Acknowledgments 5

1 Introduction 10

2 Real Gases 11

2.1 Calorically Imperfect Gases 11

2.2 Non-Ideal Gases 12

3 Solving the Euler Equations 15

3.1 Calorically Imperfect Gas 16

3.2 Non-Ideal Gas 17

4 Results 20

4.1 One Dimensional Duct Flow 21

4.2 Two Dimensional Results 23

5 Conclusions 26

A Curve Fit For SF_6 State Equation 27

6 Blank

11

B MSES Subroutine for Non-Ideal Gas Model 28

Bibliography 43

12

List of Figures

4.1 Stagnation Pressure Ratio(Strength) vs. Upstream Mach No. for Air and SF_6 at 1atm and 3atm 20

4.2 One dimensional Duct Flow 21

4.3 Shock Strength and Location vs. β 22

4.4 Shock Strength and Location vs. Z_0 22

4.5 Comparison of Air and SF_6 at Fixed M and C_L 24

4.6 Comparison of SF_6 at 1atm and 3atm to Air, $M^* = .740$, $C_L = .9$ . . . 24

4.7 Comparison of SF_6 at 1atm and 3atm to Air, $M^* = .732$, $C_L = .75$ . . 25

4.8 Comparison of SF_6 at 1atm and 3atm to Air, $\kappa = .439$, $AC_L = 2.18$ . . 25

Chapter 1
Introduction

In the past few decades, the design and development of large transport aircraft has relied on wind tunnel data taken at significantly lower Reynolds numbers than those found in operation. The drawbacks of this subscale data become apparent when one considers phenomena such as attachment line transition or similar aspects of boundary layer behavior at high Reynolds numbers.

The need for accurate wind tunnel data clearly mandates the construction of a suitable high Reynolds number test facility. However, the cost of building a large atmospheric tunnel and large tunnel models is prohibitive. Higher Reynolds numbers are often achieved by pressurizing tunnels to effectively increase the density of the air. This alternative is practical only up to a point.

A potential solution following the same basic idea relies upon the use of gases with significantly higher molecular weights than air. Candidate gases include Freon-12 or Sulfur Hexaflouride (SF_6), but the use of non-breathable gases clearly causes some problems. These problems will likely be insignificant to the cost and operational advantages of such a facility. Combining heavy gases with pressurization would allow test Reynolds numbers comparable to those on large transports in flight [1].

One complication is that Freon and SF_6 have significantly different thermodynamic properties than air, especially at elevated pressures. Heavy gases do not follow the ideal equation of state $P = \rho RT$ nearly as well as air does, nor do they maintain a constant ratio of specific heats $\gamma \equiv c_p/c_v$ over any significant temperature range. The following discussion will attempt to quantify the potential importance of these effects through a computational study.

Chapter 2
Real Gases

The thermodynamic relations specifically subject to real gas effects are the state equation

$$p = \rho RT \tag{2.1}$$

and the caloric equation,

$$h \equiv \int c_p \, dT = c_p T \tag{2.2}$$

these particular forms only being valid for a perfect gas. Real gas effects may be divided into two categories:

1. Calorically imperfect gases for which c_p depends on temperature, but which still satisfy equation (2.1).

2. Non-ideal gases for which c_p depends on both pressure and temperature, and equation (2.1) no longer holds.

The first effect results from the introduction of multiple vibrational modes for polyatomic molecules which become more important at higher temperatures. The second effect depends on intermolecular forces which become stronger as a gas moves towards liquefaction, ie. higher pressures and lower temperatures.

2.1 Calorically Imperfect Gases

The only difference between a perfect and an imperfect gas stems from the dependence of c_p on temperature in the imperfect case. A cursory examination of experimental data for SF_6 shows that, in the range of temperatures likely to be found in a wind tunnel

test, this dependence is linear in temperature.

$$c_p(T) = a + bT \tag{2.3}$$

Therefore, equation (2.2) becomes

$$h(T) = aT + \frac{bT^2}{2} \tag{2.4}$$

which may be easily inverted to find $T(h)$.

$$T(h) = -\frac{a}{b} + \sqrt{\left(\frac{a}{b}\right)^2 + \frac{2h}{b}} \tag{2.5}$$

2.2 Non-Ideal Gases

The state equation for a perfect gas (2.1) derives from a kinetic model of gas molecules which assumes that the molecules are point masses and that they do not exert any forces on one another except instantaneously during collisions. Clearly these assumptions become less accurate as the molecular weight of the gas increases. Van der Waals's equation

$$\left(p + \rho^2\alpha\right)(1 - \rho\beta) = \rho RT \tag{2.6}$$

contains two correction to equation (2.1): α corrects the pressure to account for intermolecular attraction, and β corrects for the volume of the molecules themselves.

Using a non-ideal state equation like Van der Waals's causes many serious complications as enthalpy, c_p, γ, etc. now depend on pressure as well as temperature. Despite these complications, enthalpy and entropy must remain *state variables* regardless of the form of the state equation. That is, local entropy and enthalpy must depend only on the local pressure and temperature and *not* on the upstream conditions (ie. the gas history).

Liepmann and Roshko [2] equate this condition with the requirement that a canonical equation of state must have one of these four forms:

$$e = e(s, \rho) \tag{2.7}$$

$$h = h(s,p) \tag{2.8}$$

$$f = f(T,\rho) \tag{2.9}$$

$$g = g(T,p) \tag{2.10}$$

Here $e = h - p/\rho$ is the usual internal energy, $f \equiv e - Ts$ is the free energy, and $g \equiv h - Ts$ is the free enthalpy.

For a conventional flow solver, the enthalpy defintion (2.8) appears best; however, specifying the state in this specific form is not convenient because the entropy s is not readily available to the flow solver. Liepmann and Roshko propose a more suitable form

$$\frac{p}{\rho RT} = Z(p,T) \tag{2.11}$$

which requires $T(\rho,h)$ to have a form which makes h a state variable.

For a Van der Waals's gas

$$Z = \frac{1}{1-\beta\rho} - \frac{\alpha\rho}{RT} \tag{2.12}$$

which clearly approaches the ideal state equation for $\alpha,\beta \to 0$. For typically small values of α and β

$$Z \simeq 1 + \rho\left(\beta - \frac{\alpha}{RT}\right) \simeq 1 + \frac{p}{RT}\left(\beta - \frac{\alpha}{RT}\right) \tag{2.13}$$

where the second approximation is made to make $Z = Z(p,T)$ explicitly. Liepmann and Roshko write equation (2.13) in more general form as

$$Z = 1 + \frac{p}{p_c}\phi\left(\frac{T_c}{T}\right) \tag{2.14}$$

with p_c and T_c being the critical pressure and temperature of the gas, and ϕ evidently being a universal function which they tabulate for gases other than air but with approximately the same molecular weight. For heavier gases such as SF_6 it is best to fit a curve to experimental data as explained in Appendix A. For SF_6, a good curve fit takes the form

$$\phi(\frac{T_c}{T}) = c_2\left(\frac{T_c}{T}\right)^2 + c_1\left(\frac{T_c}{T}\right) + c_0 \tag{2.15}$$

It is now necessary to determine the specific heat capacity $c_p(p,T)$ so that the enthalpy function $h(p,T)$ can be obtained. Liepmann and Roshko combine two forms of the

equation of state $h(p,T)$ and $s(p,T)$ into the fundamental reciprocity relation between $h(p,T)$ and $\rho(p,T)$

$$\frac{\partial h}{\partial p} = \frac{1}{\rho} - T\frac{\partial(1/\rho)}{\partial T} \tag{2.16}$$

which is valid for any gas. Combining this with the state equation (2.11) gives

$$\frac{\partial h}{\partial p} = -\frac{RT^2}{p}\left(\frac{\partial Z}{\partial T}\right)_p = \frac{RT_c}{p_c}\phi'\left(\frac{T_c}{T}\right) \equiv \mathcal{F}(T) \tag{2.17}$$

Since $\partial h/\partial p = \mathcal{F}(T)$ only depends on the temperature, both h and c_p must be linear in the pressure as follows.

$$h(p,T) = \int \bar{c_p}(T)\,dT + p\,\mathcal{F}(T) \tag{2.18}$$

$$c_p(p,T) \equiv \frac{\partial h}{\partial T} \tag{2.19}$$

$$= \bar{c_p}(T) + p\frac{d\mathcal{F}}{dT} \tag{2.20}$$

$$= \bar{c_p}(T) - R\frac{p}{p_c}\frac{T_c^2}{T^2}\phi''\left(\frac{T_c}{T}\right) \tag{2.21}$$

As in the case of the calorically imperfect gas, $\bar{c_p}(T)$ has the form

$$\bar{c_p}(T) = a + bT \tag{2.22}$$

Substituting this into the enthalpy equation gives

$$h(p,T) = aT + \frac{bT^2}{2} + \frac{pRT_c}{p_c}\phi'(\frac{T_c}{T}) \tag{2.23}$$

It is also possible to determine the caloric equation by expressing the internal energy (e) as $e(\rho,T)$ [3].

Chapter 3
Solving the Euler Equations

These gas models may be readily integrated into an existing flow solver which solves the integral form of the steady Euler equations:

$$\oint \rho \vec{u} \cdot \hat{n}\, dA = 0 \tag{3.1}$$

$$\oint (\rho \vec{u} \cdot \hat{n}\, \vec{u} + p\hat{n})\, dA = 0 \tag{3.2}$$

$$h_0 \equiv h + \frac{|\vec{u}|^2}{2} = \text{constant} \tag{3.3}$$

These equations are exact for any fluid flow, but must be supplied with a state equation to relate the pressure p to the enthalpy h and the density ρ. In addition, the upwinding scheme used to capture the shocks requires the local Mach number while the boundary conditions and evaluation of shock losses require the local stagnation conditions.

It is desirable to nondimensionalize the equations, and the following scheme is used where $\hat{()}$ denotes the dimensional quantitiy and $()_{ref}$ denotes a reference quantity:

$$\begin{aligned} p &= \hat{p}/p_{\text{ref}} \\ \rho &= \hat{\rho}/\rho_{\text{ref}} \\ T &= \hat{T}/T_{\text{ref}} \\ h &= \hat{h}\frac{\rho_{\text{ref}}}{p_{\text{ref}}} \end{aligned}$$

Furthermore, c_p, c_v, and R are nondimensionalized using R resulting in several new nondimensional parameters.

$$\begin{aligned} \alpha &= a/R \\ \beta &= \frac{bT_{\text{ref}}}{2a} \\ \pi &= P_{\text{ref}}/P_c \\ \tau &= T_{\text{ref}}/T_c \end{aligned}$$

For the results presented here, the reference conditions are chosen to be stagnation conditions.

3.1 Calorically Imperfect Gas

The nondimensional form of the caloric equation which governs the behavior of the imperfect gas is:

$$h(T) = \int c_p \, dT \tag{3.4}$$

$$= \alpha T + \alpha\beta T^2 \tag{3.5}$$

which may be inverted to give T as a function of h.

$$T(h) = \frac{-1 + \sqrt{1 + 4\beta h/\alpha}}{2\beta} \tag{3.6}$$

With T obtained from h, p may be determined using the ideal gas law (2.1) and a specified value of ρ. The local Mach number comes from the familiar defintion of the speed of sound:

$$a^2 \equiv \left.\frac{\partial p}{\partial \rho}\right|_s = \gamma T \tag{3.7}$$

The local value of γ may be found from equation (2.3).

$$\gamma = \frac{c_p}{c_v} = \frac{\alpha + 2\alpha\beta T}{1 - \alpha - 2\alpha\beta T} \tag{3.8}$$

The last remaining difficulty is the determination of the isentropic relations between pressure, density, and temperature. These relations are necessary to calculate stagnation conditions from flow conditions. The familiar perfect gas relations

$$\frac{T}{T_0} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{-1} \qquad \frac{\rho}{\rho_0} = \left(\frac{T}{T_0}\right)^{\frac{1}{\gamma - 1}} \qquad \frac{p}{p_0} = \left(\frac{T}{T_0}\right)^{\frac{\gamma}{\gamma - 1}}$$

do not hold for a calorically imperfect gas.

The proper forms are obtained from the formal statement,

$$dh = T\,ds + \frac{dp}{\rho} \tag{3.9}$$

and for an isentropic process $ds = 0$:

$$dh = \frac{dp}{\rho} \tag{3.10}$$

From the definition of enthalpy $dh = c_p\, dT$, and for an ideal gas $p/\rho = T$, so equation (3.10) becomes

$$\frac{c_p(T)\, dT}{T} = \frac{dp}{p} \tag{3.11}$$

Integrating this equation gives

$$\frac{p_0}{p} = \exp(-\alpha \log T + 2\alpha\beta(1 - T)) \tag{3.12}$$

and the isentropic density relation then follows directly from the state equation.

$$\frac{\rho}{\rho_0} = \frac{p}{p_0}\frac{T(h_0)}{T(h)} \tag{3.13}$$

Strictly speaking, solution of the Euler equations requires nothing else. However, if a Newton-Raphson technique is used, all of the necessary equations must be linearized for the Jacobian matrix. In the case of the calorically imperfect gas, the equations are slightly more complicated than for a perfect gas, but they may still all be written explicitly. Therefore the linearizations are easily done by differentiating the relevant equations.

3.2 Non-Ideal Gas

The nondimensional equations describing the non-ideal gas are the state equation

$$\frac{p}{\rho T} = \frac{1 + p\pi\phi(\frac{1}{\tau T})}{Z_0} \tag{3.14}$$

and the caloric equation.

$$h(p,T) = \left[\alpha T + \alpha\beta T^2 + p\frac{\pi}{\tau}\phi'(\frac{1}{\tau T})\right]\frac{1}{Z_0} \tag{3.15}$$

Z_0 is another parameter which may be described in terms of π and τ.

$$Z_0 = \frac{p_0}{\rho_0 T_0} = 1 + p_0\pi\phi(\frac{1}{\tau T_0}) \tag{3.16}$$

The non-ideal gas presents some difficulty as the enthalpy depends on the temperature *and* the pressure. Therefore, from equations (3.14) and (3.15), p and T may be found using a Newton-Raphson system to drive the following residuals to zero.

$$R_1(p,T) = \frac{p}{\rho T} - \frac{1 + p\pi\phi(\frac{1}{\tau T})}{Z_0} \tag{3.17}$$

$$R_2(p,T) = h - \left[\alpha T + \alpha\beta T^2 + p\frac{\pi}{\tau}\phi'(\frac{1}{\tau T})\right]\frac{1}{Z_0} \tag{3.18}$$

The local Mach number depends on the speed of sound which must be found from the definition:

$$a^2 = \left.\frac{\partial p}{\partial \rho}\right|_s \tag{3.19}$$

This is calculated as follows:

$$dp = \left.\frac{\partial p}{\partial \rho}\right|_h d\rho + \left.\frac{\partial p}{\partial h}\right|_\rho dh \tag{3.20}$$

but $dh = dp/\rho$ for an isentropic process, and hence

$$a^2 = \left.\frac{\partial p}{\partial \rho}\right|_s = \frac{\frac{\partial p}{\partial \rho}|_h}{1 - \frac{\partial p}{\partial h}|_\rho \frac{1}{\rho}} \tag{3.21}$$

The local γ really has no meaning and need not be calculated.

The extra complexity of the non-ideal gas appears in the calculation of the sensitivities. Since p and T are found by an iterative process they must be found by perturbing the Jacobian matrix of the converged Newton-Raphson system. A perturbation in h and ρ is related to a perturbation in p and T by the condition that the $R(p,T,h,\rho)$ must remain zero.

$$\begin{Bmatrix} \delta R_1 \\ \delta R_2 \end{Bmatrix} = 0 = \begin{bmatrix} \frac{\partial R_1}{\partial h} & \frac{\partial R_1}{\partial \rho} \\ \frac{\partial R_2}{\partial h} & \frac{\partial R_2}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} + \begin{bmatrix} \frac{\partial R_1}{\partial p} & \frac{\partial R_1}{\partial T} \\ \frac{\partial R_2}{\partial p} & \frac{\partial R_2}{\partial T} \end{bmatrix} \begin{Bmatrix} \delta p \\ \delta T \end{Bmatrix} \tag{3.22}$$

Numerically inverting this system gives the required derivatives.

$$\begin{Bmatrix} \delta p \\ \delta T \end{Bmatrix} = \begin{bmatrix} \frac{\partial p}{\partial h} & \frac{\partial p}{\partial \rho} \\ \frac{\partial T}{\partial h} & \frac{\partial T}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} \tag{3.23}$$

The second derivatives are found in a similar fashion starting instead with $\frac{\partial R_{1h}}{\partial h}$ and $\frac{\partial R_{1h}}{\partial \rho}$ as the residuals. Using a subscript notation for the derivatives ($\frac{\partial p}{\partial h} \equiv p_h$):

$$\begin{Bmatrix} \delta R_{1h} \\ \delta R_{2h} \end{Bmatrix} = 0 = \begin{bmatrix} \frac{\partial R_{1h}}{\partial h} & \frac{\partial R_{1h}}{\partial \rho} \\ \frac{\partial R_{2h}}{\partial h} & \frac{\partial R_{2h}}{\partial \rho} \end{bmatrix} \begin{Bmatrix} \delta h \\ \delta \rho \end{Bmatrix} + \begin{bmatrix} \frac{\partial R_{1h}}{\partial p_h} & \frac{\partial R_{1h}}{\partial T_h} \\ \frac{\partial R_{2h}}{\partial p_h} & \frac{\partial R_{2h}}{\partial T_h} \end{bmatrix} \begin{Bmatrix} \delta p_h \\ \delta T_h \end{Bmatrix} \tag{3.24}$$

A similar system with $R_{1\rho}$ and $R_{2\rho}$ as residuals is also formed. As above, numerically inverting gives $\frac{\partial^2 p}{\partial h^2} = \frac{\partial p_h}{\partial h}$, $\frac{\partial^2 p}{\partial h \partial \rho} = \frac{\partial p_h}{\partial \rho}$, etc. These manipulations are implemented in the source code in Appendix B.

The last remaining task is calculation of the stagnation conditions and, again, it is not possible to find an analytic expression. Another Newton-Raphson system is constructed where the first residual comes from equation(3.15):

$$R_1 = h_0 - h(p, T) \tag{3.25}$$

The second residual is derived by rearranging equation(3.9)

$$ds = \frac{dh}{T} + \frac{dp}{\rho T} \tag{3.26}$$

$$= \frac{\bar{c_p}}{T} dT + \frac{d(p\mathcal{F})}{T} - \frac{dp}{p} Z \tag{3.27}$$

$$= \frac{\bar{c_p}}{T} dT + d(p\pi \frac{1}{\tau T} \phi') - \pi d(p\phi) - \frac{dp}{p} \tag{3.28}$$

Integrating gives:

$$s(p, T) = \int \frac{\bar{c_p}}{T} dT + p\pi \left[\frac{1}{\tau T} \phi' - \phi \right] - ln(p) \tag{3.29}$$

The second residual may then be formed

$$R_2 = s_1 - s(p, T) \tag{3.30}$$

where s_1 is the entropy of the static conditions.

Driving these two residuals to zero gives the stagnation conditions p_0, T_0. The derivatives $\frac{\partial p_0}{\partial \rho}$, $\frac{\partial p_0}{\partial h}$, etc, needed for the Newton-Raphson solver may then be found by perturbing the converged Jacobian matrix and relating the resulting derivatives to the static conditions through the chain rule and equations (3.15) and (3.29). This process is identical to the one used above to find p and T and their derivatives.

Chapter 4

Results

After developing the models for the calorically imperfect and non-ideal gases, the next step was to evaluate the differences these changes caused in inviscid flows. The primary quantities of interest are the location of shocks and their strength which is defined as the ratio of of stagnation pressures across the shock. For a perfect gas, the shock strength may be expressed as a function of the upstream Mach number M_1.

$$\frac{p_{02}}{p_{01}} = \left[1 + \frac{2\gamma}{\gamma+1}(M_1^2 - 1)\right]^{-1/(\gamma-1)} \left[\frac{(\gamma+1)M_1^2}{(\gamma-1)M_1^2 + 2}\right]^{\gamma/(\gamma-1)} \tag{4.1}$$

However, for the non-ideal gas, this relation must be calculated numerically.

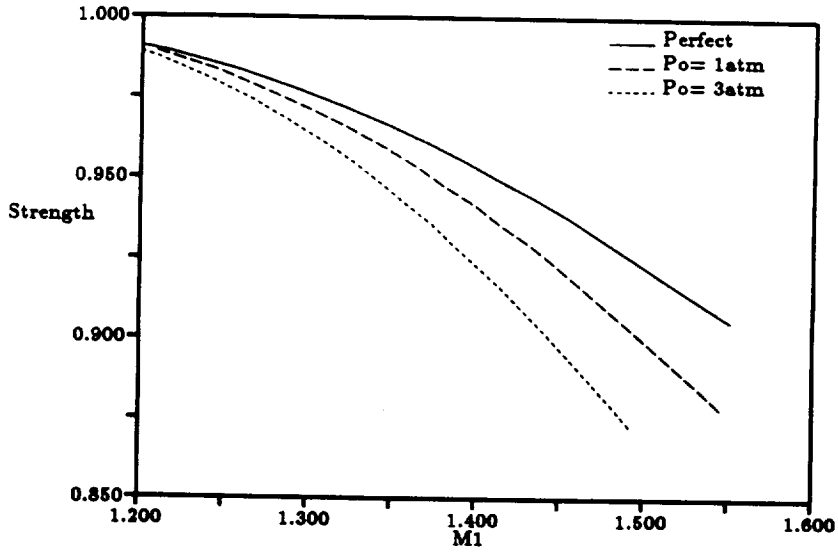

Figure 4.1: Stagnation Pressure Ratio(Strength) vs. Upstream Mach No. for Air and SF_6 at 1atm and 3atm

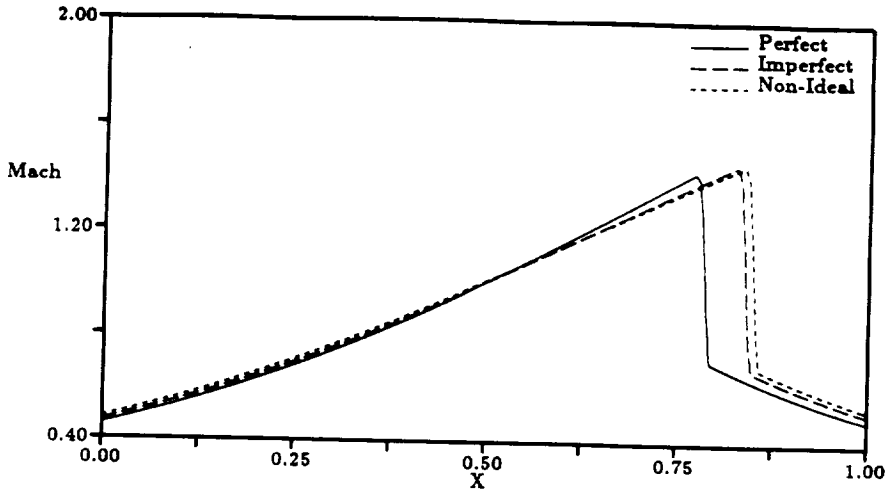

Figure 4.2: One dimensional Duct Flow

4.1 One Dimensional Duct Flow

The first comparison of the different gas models was a study of the flow in a converging/diverging nozzle using a quasi one dimensional Euler solver. This flow is characterized by sonic flow at the throat with a shock downstream to match the specified exit pressure as shown in figure(4.2).

As a basis for comparison of the different gas models in a duct flow, the non-dimensional reference enthalpy ($h_0\rho_0/p_0$) was made equal for all three cases.

$$h_0 = \frac{\gamma}{\gamma - 1} \tag{4.2}$$

$$= \alpha(1+\beta) \tag{4.3}$$

$$= \frac{\alpha(1+\beta) + \pi\phi'(\frac{1}{\tau})}{Z_0} \tag{4.4}$$

With h_0 held constant, γ therefore depends on α, β, π, and τ. The exit presure ratio is also held constant. Under these conditions, the slope of the c_p versus T curve (β) had little or no effect on shock strength or position relative to the perfect gas as shown in figure(4.3).

25

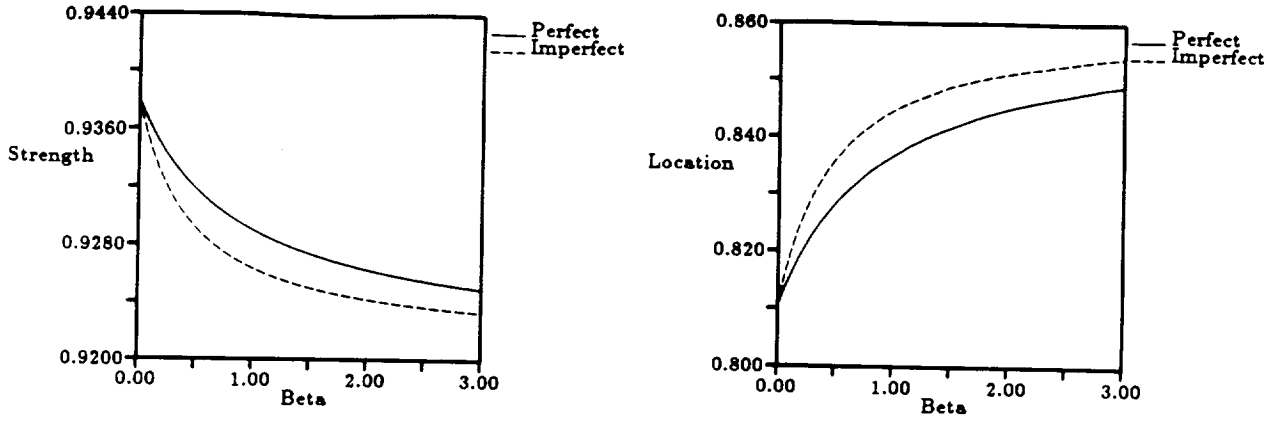

Figure 4.3: Shock Strength and Location vs. β

For the non-ideal gas, π and τ are not really independent parameters and may be combined into Z_0. Figure(4.4) shows the variation in shock strength and position as functions of Z_0 and the corresponding perfect gas results with γ adjusted to preserve the stagnation enthalpy as above. These plots clearly show that it is not possible to mimic the effects of the non-ideality by changing γ as in the case of the calorically imperfect gas. The difference in shock strength and position becomes larger and larger as the gas becomes less ideal.

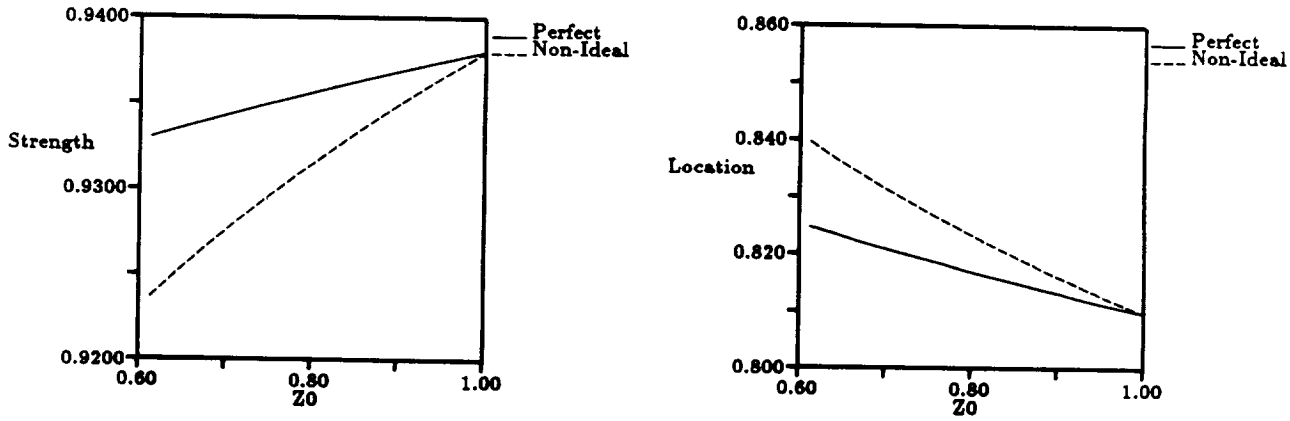

Figure 4.4: Shock Strength and Location vs. Z_0

The last test conducted with the one dimensional flow model was to determine the effects of the various gas models on the upwinding scheme needed for stability of the numerical scheme. The flow solver drives the momentum equation residual to zero,

$$R_1 \equiv \rho_i q_i A_i(\bar{q}_i - q_{i-1}) + p_i A_i - p_{i-1} A_{i-1} + \frac{p_i + p_{i-1}}{2}(A_i - A_{i-1}) \qquad (4.5)$$

where the upwinded speed is defined as

$$\bar{q}_i = q_i - \mu_i(q_i - q_{i-1}) \qquad (4.6)$$

and μ_i is non-zero only if M_i is greater than M_c.

$$\mu_i(M_i(q_i)) = \frac{K_\mu}{\gamma}\left[1 - \frac{M_c^2}{M_i^2}\right] \qquad (4.7)$$

Initially, the exact γ was calculated at each node along with all the necessary linearizations and used in the upwinding scheme. Under these conditions, the flow solver converged with $M_c \leq 1$. However, the upwinding is relatively insensitive to the exact value of γ even though the stability analysis used to derive equation(4.7) ignored γ perturbations. Using a constant value of γ had absolutely no effect on the viable range for M_c or the rate of convergence.

4.2 Two Dimensional Results

The subroutine which appears in Appendix B was incorporated into MSES, the multi-element version of the two dimensional transonic airfoil design/analysis code ISES [4]. Numerical experiments carried out were limited to single-element inviscid cases to more clearly demonstrate the effect of the new gas model. Figure(4.5) shows an overlay of the Mach distributions for a test airfoil run in SF_6 at two different stagnation conditions and in air. All three cases are at matched freestream Mach number and lift coefficient. Note that they are *not* at the same angle of attack. The SF_6 is characterized by stagnation pressures of 1atm and 3atm and a stagnation temperature of $310K$.

Airfoils tests in heavy gases will be much more worthwile if some relationship may be found so that the tests reflect the airfoil performance in air. The only parameters

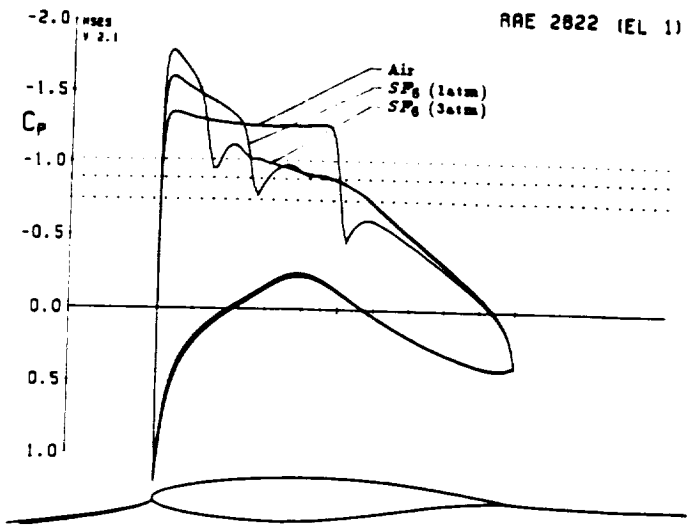

Figure 4.5: Comparison of Air and SF_6 at Fixed M and C_L

which may be adjusted in a wind tunnel test are the Mach number, stagnation conditions, and angle of atttack or C_L. Figure(4.5) shows an attempted match keeping M and C_L constant: clearly, this is not an effective technique. After a good deal of experimentatation, the best match was achieved by running the different gases at the same M^* which is defined as the ratio of freestream velocity to the speed of sound at sonic conditions. Figure(4.6) shows the case in air from figure(4.5) compared with SF_6 (1atm and 3atm) at the same M^*.

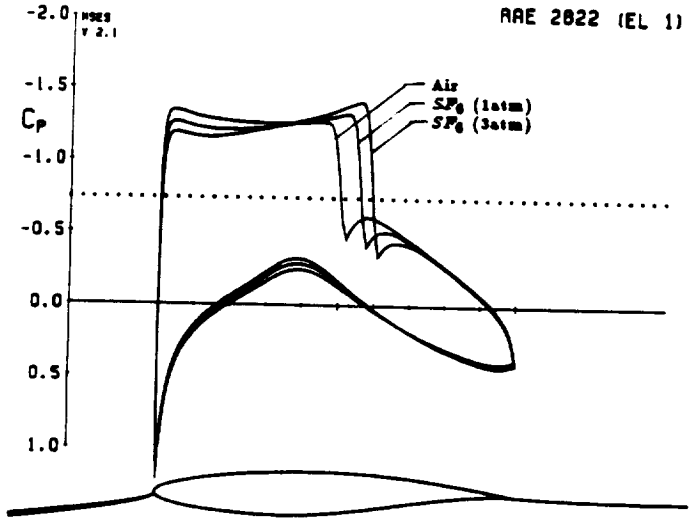

Figure 4.6: Comparison of SF_6 at 1atm and 3atm to Air, $M^* = .740$, $C_L = .9$

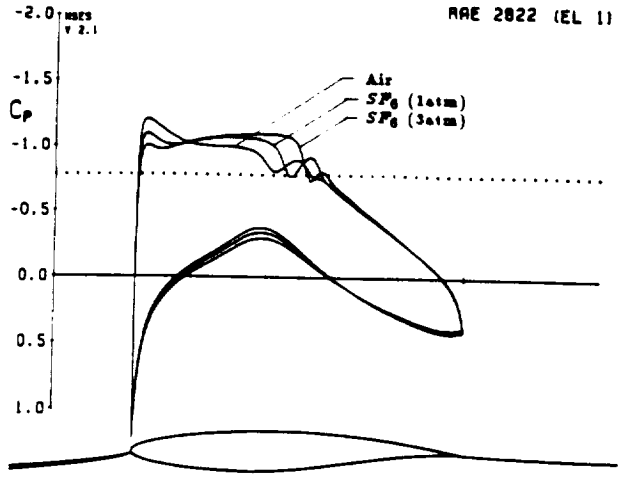

Figure 4.7: Comparison of SF_6 at 1atm and 3atm to Air, $M^* = .732$, $C_L = .75$

A case with a weaker shock, figure(4.7) was used to further verify this relationship. The match is slightly worse, but this is to be expected because a weak shock is much more sensitive to small changes in M than a strong one. As an alternative to matching M^*, Anderson [5] proposes matching the small disturbance similarity parameter κ and AC_L where

$$\kappa = \frac{1 - M_\infty^2}{(M_\infty^2(\gamma' + 1))^{2/3}} \tag{4.8}$$

$$A = \frac{M_\infty^2(\gamma' + 1)}{1 - M_\infty^2} \tag{4.9}$$

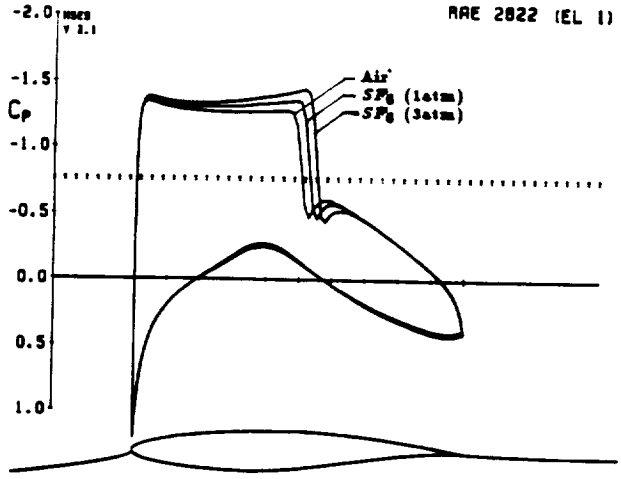

Figure 4.8: Comparison of SF_6 at 1atm and 3atm to Air, $\kappa = .439$, $AC_L = 2.18$

24

Chapter 5
Conclusions

The models derived above adequately describe the thermodynamic behavior of non-ideal and calorically imperfect gases. Despite some minor complications in linearizing these models, they were implemented in routines suitable for incorporation into existing flow solvers based on Newton's method. First, a quasi one-dimensional flow solver was used to examine the influence of the various non-dimensional parameters which govern the behavior of the different gases.

Transonic airfoil test cases for air and SF_6 were then used to study the influence of parameters which may be controlled in a wind tunnel experiment: stagnation pressure, freestream Mach number, and angle of attack. The goal of this study was determine the conditions under which a wind tunnel test in a heavy gas would produce results comparable to those found in air. Matching M^* and C_L or κ and AC_L were both effective for the test cases presented here. Further study is necessary to determine which is best for multi-element cases.

The results are encouraging in that they definitely hint at the possibility of directly relating heavy gas test data to performance in air. It is first necessary to verify experimentally the model for SF_6, and to investigate the effects of non-ideal gases on viscous flows.

Appendix A

Curve Fit For SF_6 State Equation

A curve fit may be found for the function $\phi\left(\frac{T_{ref}}{T}\right)$ for any gas given experimental state data. With the density (ρ) measured at a number of different pressures (p) and temperatures (T), a vector is defined containing the difference between the real gas and a perfect gas at each data point.

$$\vec{Z} = \begin{bmatrix} \frac{p_1}{\rho_1 R T_1} - 1 \\ \vdots \\ \frac{p_m}{\rho_m R T_m} - 1 \end{bmatrix} \tag{A.1}$$

Defining $\theta \equiv \frac{T_{ref}}{T}$, the matrix A contains the state information.

$$\bar{A} = \frac{1}{p_{ref}} \begin{bmatrix} p_1\theta_1^n & p_1\theta_1^{n-1} & \cdots & p_1\theta_1^2 & p_1\theta_1 & p_1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ p_m\theta_m^n & p_m\theta_m^{n-1} & \cdots & p_m\theta_m^2 & p_m\theta_m & p_m \end{bmatrix} \tag{A.2}$$

The goal is to find a state equation agreeing closely with the experimental data in $\vec{Z}$ but of the simple form:

$$Z(p,T) = 1 + \frac{p}{p_{ref}} \begin{bmatrix} C_n & C_{n-1} & \cdots & C_0 \end{bmatrix} \begin{bmatrix} \theta^n \\ \theta^{n-1} \\ \vdots \\ 1 \end{bmatrix} \tag{A.3}$$

Therefore

$$\vec{Z} \simeq A\vec{C} \tag{A.4}$$

and $\vec{C}$ is found by the technique of linear regression:

$$\vec{C} = (A^T A)^{-1} A^T Z \tag{A.5}$$

The results presented in this thesis were based on a quadratic fit for ϕ from approximate data for SF_6. The required data may be found in [6].

Appendix B

MSES Subroutine for Non-Ideal Gas Model

```
      subroutine hgparm(alf1,bta1, tau1, cc0,cc1,cc2, h0)
c-----------------------------------------------------------------------
c     Initializes non-ideal gas routines.
c     Formulation derived in Schafer SM thesis.
c
c     Input:
c      alf1  Constants for Cp(T) in caloric equation:  Cp = a(1 + bT)
c      bet1
c
c      tau1  Constant in phi(T) in non-ideality factor Z(p,T)
c
c      cc0   Constants defining phi(T) in polynomial form:
c      cc1
c      cc2         phi = c0 + c1(tau/T) + c2(tau/T)**2
c
c     Output:
c      h0    Enthalpy at reference conditions  p0, T0
c
c     Internal output:
c      z0    Non-ideality factor Z(p0,T0) at reference conditions
c
c-----------------------------------------------------------------------
      implicit real*4 (a-h,m,o-z)
      common /nongas/
     &          alf, bta, pi, tau, z0
      common /nonfit/
     &          c2, c1, c0
c
c---- put input parameters into common blocks
      alf = alf1
      bta = bta1
c
      tau = tau1
c
      c0 = cc0
      c1 = cc1
      c2 = cc2
c
      pi = 1.0
c
c---- calculate reference non-ideality factor and enthalpy
      z0 = 1.0 + pi*(c2/tau**2 + c1/tau + c0)
      h0 = (alf*(1. + bta) + pi/tau*phid(1./tau)) / z0
c
```

```
      return
      end



      subroutine nideal(h0,r,q, p  ,p_r  ,p_q,
     &                           msq,msq_r,msq_q)
c-----------------------------------------------------
c     Calculates pressure and Mach number for specified
c     stagnation enthalpy, density, and speed.
c
c     Input:
c      h0     stagnation enthalpy
c      r      density
c      q      speed
c
c     Output:
c      p      pressure
c      p_r    dp/dr
c      p_q    dp/dq
c      msq    square of Mach number M^2
c      msq_r  dM^2/dr
c      msq_q  dM^2/dq
c-----------------------------------------------------
      implicit real*4 (a-h,m,o-z)
c
c---- set static enthalpy
      h   = h0 - 0.5*q**2
      h_q =         -q
c
c---- set pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
     &               t,t_r,t_h,t_rr,t_hh,t_rh)
      p_q = p_h*h_q
c
c---- set speed of sound squared:  a^2 = dp/dr  (at constant s)
      asq   = p_r  / (1. - p_h/r)
      asq_r = p_rr / (1. - p_h/r)
     &      - p_r  / (1. - p_h/r)**2 *(p_h/r**2 - p_rh/r)
      asq_h = p_rh / (1. - p_h/r)
     &      + p_r  / (1. - p_h/r)**2 *p_hh/r
      asq_q = asq_h*h_q
c
c---- set Mach number squared
      msq   = q**2/asq
      msq_r = -msq/asq * asq_r
      msq_q = -msq/asq * asq_q  +  2.*q/asq
c
      return
      end



      subroutine ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
```

```
     &                          t,t_r,t_h,t_rr,t_hh,t_rh)
c-------------------------------------------------------
c     Calculates pressure and temperature for
c     specified static enthalpy and density.
c
c     Input:
c      h       enthalpy
c      r       density
c
c     Output:
c      p       pressure
c      p_r     dp/dr
c      p_h     dp/dh
c      p_rr    d^2p/dr^2
c      p_hh    d^2p/dh^2
c      p_rh    d^2p/drdh
c      t       temperature
c      t_r     dt/dr ... etc.
c-------------------------------------------------------
      implicit real*4 (a-h,m,o-z)
      dimension a(2,2), ai(2,2), aih(2,2), air(2,2),
     &  b(2,2), bh(2,2), br(2,2)
      common /nongas/
     &          alf, bta, pi, tau, z0
c
c---- Newton convergence tolerance
      data eps /5.0E-6/
c
c---- initial guess from imperfect ideal gas
      if(bta.eq.0.0) then
       t = h/alf
      else
       t  = (-1.0 +  sqrt(1.0 + 4.0*bta*h/alf)) / (2.0*bta)
      endif
      p = r*t
c
c---- Newton loop to converge on correct p,t
      itcon = 15
      do 100 iter=1, itcon
c
c---- set and linearize non-ideality factor Z(p,t)
      ttc   =  1./(tau*t)
      ttc_t = -1./(tau*t**2)
c
      z   = 1. + p*pi*phi(ttc)
      z_p =        pi*phi(ttc)
      z_t =      p*pi*phid(ttc)*ttc_t
c
c---- residual 1: state equation
      res1 =  p/(r*t)    - z  /z0
      r1_p = 1./(r*t)    - z_p/z0
      r1_t = -p/(r*t**2) - z_t/z0
c
c
```

```
      tm1   = (alf*t +    alf*bta*t**2) / z0
      tm1_p = 0.
      tm1_t = (alf   + 2.*alf*bta*t   ) / z0
c
      tm2   = p*pi/tau*phid(ttc)         / z0
      tm2_p =   pi/tau*phid(ttc)         / z0
      tm2_t = p*pi/tau*phidd(ttc)*ttc_t  / z0
c
c---- residual 2: caloric equation
      res2 = h - (tm1   + tm2)
      r2_p =   - (tm1_p + tm2_p)
      r2_t =   - (tm1_t + tm2_t)
c
c---- set Jacobian matrix
      a(1,1)  = r1_t
      a(1,2)  = r1_p
      a(2,1)  = r2_t
      a(2,2)  = r2_p
c
c---- find inverse Jacobian matrix
      detinv  =  1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
      ai(1,1) =  a(2,2)*detinv
      ai(2,2) =  a(1,1)*detinv
      ai(1,2) = -a(1,2)*detinv
      ai(2,1) = -a(2,1)*detinv
c
c---- set Newton changes
      dt = -(ai(1,1)*res1 + ai(1,2)*res2)
      dp = -(ai(2,1)*res1 + ai(2,2)*res2)
c
      rlx = 1.0
      if(rlx*dp .gt. 2.5*p) rlx = 2.5*p/dp
      if(rlx*dp .lt. -.8*p) rlx = -.8*p/dp
      if(rlx*dt .gt. 2.5*t) rlx = 2.5*t/dt
      if(rlx*dt .lt. -.8*t) rlx = -.8*t/dt
c
c---- update variables
      t = t + rlx*dt
      p = p + rlx*dp
c
c---- convergence check
      if (abs(dp/p) .le. eps .and. abs(dt/t) .le. eps) goto 3
c
 100  continue
c
      write(*,*) 'NGASPT: Convergence failed.'
      write(*,*) 'dp dT  :', dp, dt
      write(*,*) 'p T h r:', p, t, h, r
c
 3    continue
c
c---- set residual derivatives wrt input r,h variables
      r1_r = -p/(r**2*t)
      r1_h = 0.
```

```
      r2_r = 0.
      r2_h = 1.
c
      b(1,1)  = r1_r
      b(1,2)  = r1_h
      b(2,1)  = r2_r
      b(2,2)  = r2_h
c
c---- set p,t derivatives wrt r,h
      t_r = -(ai(1,1)*b(1,1) + ai(1,2)*b(2,1))
      t_h = -(ai(1,1)*b(1,2) + ai(1,2)*b(2,2))
      p_r = -(ai(2,1)*b(1,1) + ai(2,2)*b(2,1))
      p_h = -(ai(2,1)*b(1,2) + ai(2,2)*b(2,2))
c
c
c---- set second residual derivatives wrt r,h
      ttc    =  1./(tau*t)
      ttc_t  = -1./(tau*t**2)
      ttc_tt =  2./(tau*t**3)
c
      z    = 1. + p*pi*phi(ttc)
      z_p  =        pi*phi(ttc)
      z_pt =        pi*phid(ttc)*ttc_t
      z_pp = 0.
      z_t  =      p*pi*phid(ttc)*ttc_t
      z_tt =      p*pi*(phidd(ttc)*ttc_t**2 + phid(ttc)*ttc_tt)
c
      r1    =    p/(r*t)    - z   /z0
      r1_p  =   1./(r*t)    - z_p /z0
      r1_pt =  -1./(r*t**2) - z_pt/z0
      r1_pp =               - z_pp/z0
      r1_t  =   -p/(r*t**2) - z_t /z0
      r1_tt = 2.*p/(r*t**3) - z_tt/z0
      r1_r  =   -p/(r**2*t)
      r1_h  = 0.
      r1_hp = 0.
      r1_ht = 0.
      r1_rp =  -1./(r**2*t)
      r1_rt =    p/(r**2*t**2)
      r1_rr = 2.*p/(r**3*t)
c
      tm1    = (alf*t +    alf*bta*t**2) / z0
      tm1_t  = (alf   + 2.*alf*bta*t   ) / z0
      tm1_tt = (        2.*alf*bta     ) / z0
      tm1_pt = 0.
      tm1_p  = 0.
      tm1_pp = 0.
c
      tm2    = p*pi/tau*phid(ttc)              / z0
      tm2_p  =   pi/tau*phid(ttc)              / z0
      tm2_pt =   pi/tau*phidd(ttc)*ttc_t       / z0
      tm2_pp = 0.
      tm2_t  = p*pi/tau* phidd(ttc)*ttc_t       / z0
      tm2_tt = p*pi/tau*(phiddd(ttc)*ttc_t**2 +
```

```
     &                  phidd(ttc)*ttc_tt)     / z0
c
      r2   = h - (tm1   + tm2)
      r2_p =   - (tm1_p + tm2_p)
      r2_t =   - (tm1_t + tm2_t)
      r2_h = 1.
c
c
c---- set and linearize new residuals: r1h = dr1/dh = 0, r2h = dr2/dh = 0
      ph = p_h
      th = t_h
c
      r1h    = r1_p *ph + r1_t *th + r1_h
      r1h_ph = r1_p
      r1h_th =            r1_t
      r1h_p  = r1_pp*ph + r1_pt*th + r1_hp
      r1h_t  = r1_pt*ph + r1_tt*th + r1_ht
      r1h_h  = 0.
      r1h_r = -ph/(r**2*t) + th*p/(r**2*t**2)
c
c
      r2h    = 1. - tm1_t*th  - tm1_p*ph  - tm2_t*th -  tm2_p*ph
      r2h_ph =                - tm1_p                -  tm2_p
      r2h_th =    - tm1_t                  - tm2_t
      r2h_p  =    - tm1_pt*th - tm1_pp*ph - tm2_pt*th - tm2_pp*ph
      r2h_t  =    - tm1_tt*th - tm1_pt*ph - tm2_tt*th - tm2_pt*ph
      r2h_h  = 0.
      r2h_r  = 0.
c
      a(1,1)  = r1h_th
      a(1,2)  = r1h_ph
      a(2,1)  = r2h_th
      a(2,2)  = r2h_ph
c
      detinv   =  1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
      aih(1,1) =  a(2,2)*detinv
      aih(2,2) =  a(1,1)*detinv
      aih(1,2) = -a(1,2)*detinv
      aih(2,1) = -a(2,1)*detinv
c
      dth = -(aih(1,1)*r1h + aih(1,2)*r2h)
      dph = -(aih(2,1)*r1h + aih(2,2)*r2h)
c
c      ph = ph + dph
c      th = th + dth
c
c
c---- set and linearize new residuals: r1r = dr1/dr = 0, r2r = dr2/dr = 0
      pr = p_r
      tr = t_r
c
      r1r    = r1_p *pr + r1_t *tr + r1_r
      r1r_pr = r1_p
      r1r_tr =            r1_t
```

```
      r1r_p  = r1_pp*pr + r1_pt*tr + r1_rp
      r1r_t  = r1_pt*pr + r1_tt*tr + r1_rt
      r1r_r  = r1_rp*pr + r1_rt*tr + r1_rr
      r1r_h  = 0.
c
c
      r2r    = - tm1_t *tr - tm1_p *pr - tm2_t *tr - tm2_p *pr
      r2r_pr =             - tm1_p                 - tm2_p
      r2r_tr = - tm1_t                 - tm2_t
      r2r_p  = - tm1_pt*tr - tm1_pp*pr - tm2_pt*tr - tm2_pp*pr
      r2r_t  = - tm1_tt*tr - tm1_pt*pr - tm2_tt*tr - tm2_pt*pr
      r2r_h  = 0.
      r2r_r  = 0.
c
      a(1,1)  = r1r_tr
      a(1,2)  = r1r_pr
      a(2,1)  = r2r_tr
      a(2,2)  = r2r_pr
c
      detinv   =  1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
      air(1,1) =  a(2,2)*detinv
      air(2,2) =  a(1,1)*detinv
      air(1,2) = -a(1,2)*detinv
      air(2,1) = -a(2,1)*detinv
c
      dtr = -(air(1,1)*r1r + air(1,2)*r2r)
      dpr = -(air(2,1)*r1r + air(2,2)*r2r)
c
c      pr = pr + dpr
c      tr = tr + dtr
c
c
c---- calculate responses in dt/dh and dp/dh to unit h perturbation
      dr1h = r1h_h + r1h_p*ph + r1h_t*th
      dr2h = r2h_h + r2h_p*ph + r2h_t*th
c
      dr1r = r1r_h + r1r_p*ph + r1r_t*th
      dr2r = r2r_h + r2r_p*ph + r2r_t*th
c
c
      dth = -(aih(1,1)*dr1h + aih(1,2)*dr2h)
      dph = -(aih(2,1)*dr1h + aih(2,2)*dr2h)
      thh = dth
      phh = dph
c
      dth = -(air(1,1)*dr1r + air(1,2)*dr2r)
      dph = -(air(2,1)*dr1r + air(2,2)*dr2r)
      thr = dth
      phr = dph
c
c---- calculate responses in dt/dh and dp/dh to unit r perturbation
      dr1h = r1h_r + r1h_p*pr + r1h_t*tr
      dr2h = r2h_r + r2h_p*pr + r2h_t*tr
c
```

```
      dr1r = r1r_r + r1r_p*pr + r1r_t*tr
      dr2r = r2r_r + r2r_p*pr + r2r_t*tr
c
c
      dth = -(aih(1,1)*dr1h + aih(1,2)*dr2h)
      dph = -(aih(2,1)*dr1h + aih(2,2)*dr2h)
      trh = dth
      prh = dph
c
      dth = -(air(1,1)*dr1r + air(1,2)*dr2r)
      dph = -(air(2,1)*dr1r + air(2,2)*dr2r)
      trr = dth
      prr = dph
c
c---- set final first and second derivatives wrt (r,h)
      p_r  = pr
      t_r  = tr
      p_h  = ph
      t_h  = th
      p_hh = phh
      t_hh = thh
      p_rr = prr
      t_rr = trr
      p_rh = .5*(prh+phr)
      t_rh = .5*(trh+thr)
c
      return
      end



      subroutine nonstag(h0,rho,q, p0,p0_r,p0_q,
     &                             r0,r0_r,r0_q )
c-----------------------------------------------------
c     Calculates stagnation pressure and density for
c     specified stagnation enthalpy, density, and speed.
c
c     Input:
c      h0     stagnation enthalpy
c      rho    density
c      q      speed
c
c     Output:
c      p0      stagnation pressure
c      p0_r    dp0/dr
c      p0_q    dp0/dq
c      r0      stagnation density
c      r0_r    dr0/dr
c      r0_q    dr0/dq
c-----------------------------------------------------
      implicit real*4 (a-h,m,o-z)
      dimension a(2,2), ai(2,2), b(2,2)
      real*4 h_p,h_t
c
```

```
      common /nongas/
     &          alf, bta, pi, tau, z0
      common /nonfit/
     &          c2, c1, c0
c
      data eps /5.0E-6/
c
      z(pp,tt)   = 1. + pp*pi*phi (1./(tau*tt))
      z_p(pp,tt) =         pi*phi (1./(tau*tt))
      z_t(pp,tt) =      pp*pi*phid(1./(tau*tt)) / (-tau*tt**2)
c
      h    = h0 - .5*q**2
      h_q  =    -    q
ccc   h_h0 = 1.0
c
      r = rho
c
c---- set input pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
     &               t,t_r,t_h,t_rr,t_hh,t_rh)
c
c---- set entropy s and derivatives wrt p,t
      ttc    =  1./(tau*t)
      ttc_t  = -1./(tau*t**2)
      ttc_tt =  2./(tau*t**3)
c
      ph    = phi(ttc)
      phd   = phid(ttc)
      phdd  = phidd(ttc)
      phddd = phiddd(ttc)
c
      ph_t   = phd   * ttc_t
      phd_t  = phdd  * ttc_t
      phdd_t = phddd * ttc_t
c
      s   = alf*log(t) + 2.0*alf*bta*t
     &    - p*pi*( t*phd  *ttc_t + ph  ) - log(p)
      s_p = - pi*( t*phd  *ttc_t + ph  ) - 1.0/p
      s_t = alf/t      + 2.0*alf*bta
     &    - p*pi*(   phd  *ttc_t + ph_t
     &             + t*phd_t*ttc_t
     &             + t*phd  *ttc_tt      )
c
c---- initial guess for p0,t0 from imperfect gas
cc      if(bta.eq.0.0) then
cc       t0 = h0/alf
cc      else
cc       t0  = (-1.0 +  sqrt(1.0 + 4.0*bta*h0/alf)) / (2.0*bta)
cc      endif
cc      p0  = p * exp(-alf*log(t) + alf*2.0*bta*(1.0-t))
c
      t0 = t
      p0 = p
c
```

```
c---- Newton loop to converge on correct p0,t0
      itcon = 15
      do 100 iter=1, itcon
c
      ttc     =  1./(tau*t0)
      ttc_t0  = -1./(tau*t0**2)
      ttc_tt0 =  2./(tau*t0**3)
c
      ph    = phi(ttc)
      phd   = phid(ttc)
      phdd  = phidd(ttc)
      phddd = phiddd(ttc)
c
      ph_t0   = phd   * ttc_t0
      phd_t0  = phdd  * ttc_t0
      phdd_t0 = phddd * ttc_t0
c
c---- enthalpy residual
      res1  = (alf*(t0 + bta*t0**2) + p0*pi/tau*phd   )/z0  -  h0
      r1_p0 = (                       pi/tau*phd   )/z0
      r1_t0 = (alf*(1.0+ bta*t0*2.) + p0*pi/tau*phd_t0)/z0
c
c---- entropy residual
      res2  = alf*log(t0) + 2.0*alf*bta*t0
     &      - p0*pi*( t0*phd   *ttc_t0 + ph   ) - log(p0)  -  s
      r2_p0 =  - pi*( t0*phd   *ttc_t0 + ph   ) - 1.0/p0
      r2_t0 = alf/t0      + 2.0*alf*bta
     &      - p0*pi*(    phd   *ttc_t0 + ph_t0
     &             + t0*phd_t0*ttc_t0
     &             + t0*phd   *ttc_tt0       )
c
c---- setup and invert Jacobian matrix
      a(1,1) = r1_t0
      a(1,2) = r1_p0
      a(2,1) = r2_t0
      a(2,2) = r2_p0
c
      detinv  =  1.0 / (a(1,1)*a(2,2) - a(1,2)*a(2,1))
      ai(1,1) =  a(2,2)*detinv
      ai(2,2) =  a(1,1)*detinv
      ai(1,2) = -a(1,2)*detinv
      ai(2,1) = -a(2,1)*detinv
c
c---- set Newton variables
      dt = -(ai(1,1)*res1 + ai(1,2)*res2)
      dp = -(ai(2,1)*res1 + ai(2,2)*res2)
c
      rlx = 1.0
      if(rlx*dp .gt. 2.5*p0) rlx = 2.5*p0/dp
      if(rlx*dp .lt. -.8*p0) rlx = -.8*p0/dp
      if(rlx*dt .gt. 2.5*t0) rlx = 2.5*t0/dt
      if(rlx*dt .lt. -.8*t0) rlx = -.8*t0/dt
c
c---- update variables
```

```
      p0 = p0 + rlx*dp
      t0 = t0 + rlx*dt
c
c---- convergence check
      if(abs(dp/p0) .le. eps .and. abs(dt/t0) .le. eps) go to 2
c
 100  continue
c
      write(*,*) 'NONSTAG: Convergence failure.'
      write(*,*) 'dp dT    :',dp, dt
      write(*,*) 'po To h r:',p0,t0,h,r
c
 2    continue
c
c---- set residual derivatives wrt (s,h0)
      r1_s = 0.
      r2_s = -1.0
      r1_h = -1.0
      r2_h = 0.
c
      b(1,1)  = r1_s
      b(1,2)  = r1_h
      b(2,1)  = r2_s
      b(2,2)  = r2_h
c
c---- set (t0,p0) derivatives wrt (s,h0)
      t0_s  = -(ai(1,1)*b(1,1) + ai(1,2)*b(2,1))
ccc   t0_h0 = -(ai(1,1)*b(1,2) + ai(1,2)*b(2,2))
      p0_s  = -(ai(2,1)*b(1,1) + ai(2,2)*b(2,1))
ccc   p0_h0 = -(ai(2,1)*b(1,2) + ai(2,2)*b(2,2))
c
c---- convert derivatives wrt (s,h0) to wrt (p,t,h0)
      t0_t = t0_s*s_t
      t0_p = t0_s*s_p
      p0_t = p0_s*s_t
      p0_p = p0_s*s_p
c
c
c---- set stagnation density r0 and derivatives wrt (p0,t0)
      zz   = z(p0,t0)
      zz_p = z_p(p0,t0)
      zz_t = z_t(p0,t0)
c
      r0   =  z0/zz      * p0/t0
      r0_z = -z0/zz**2   * p0/t0
c
      r0_p0 = r0_z*zz_p + z0/(zz*t0)
      r0_t0 = r0_z*zz_t - z0*p0/(zz*t0**2)
c
c---- convert derivatives from wrt (p0,t0) to wrt (p,t,h0)
      r0_p  = r0_p0*p0_p  + r0_t0*t0_p
      r0_t  = r0_p0*p0_t  + r0_t0*t0_t
ccc   r0_h0 = r0_p0*p0_h0 + r0_t0*t0_h0
c
```

```
c---- convert derivatives from wrt (p,t) to wrt (r,q,h0)
      r0_r =  r0_p*p_r + r0_t*t_r
      r0_q = (r0_p*p_h + r0_t*t_h)*h_q
      p0_r =  p0_p*p_r + p0_t*t_r
      p0_q = (p0_p*p_h + p0_t*t_h)*h_q
c
ccc   r0_h0 = (r0_p*p_h + r0_t*t_h)*h_h0  +  r0_h0
ccc   p0_h0 = (p0_p*p_h + p0_t*t_h)*h_h0  +  p0_h0
c
      return
      end


      real*4 function phi(ttc)
      implicit real*4(a-h,m,o-z)
c-------------------------------------------------------------
c     Returns function phi used in non-ideality parameter
c     Z = 1 + pi*phi(ttc)
c-------------------------------------------------------------
      common /nonfit/
     &          c2, c1, c0
c
      phi = c2*ttc**2 + c1*ttc + c0
c
      return
      end


      real*4 function phid(ttc)
      implicit real*4(a-h,m,o-z)
      common /nonfit/
     &          c2, c1, c0
c
      phid = 2.*c2*ttc + c1
c
      return
      end


      real*4 function phidd(ttc)
      implicit real*4(a-h,m,o-z)
      common /nonfit/
     &          c2, c1, c0
c
      phidd = 2.*c2
c
      return
      end

      real*4 function phiddd(ttc)
      implicit real*4(a-h,m,o-z)
      common /nonfit/
     &          c2, c1, c0
c
```

```
      phiddd = 0.
c
      return
      end


      subroutine hgent(h0,r,q, s)
c-------------------------------------------------
c     Returns entropy s from input variables h0,r,q
c-------------------------------------------------
      common /nongas/
     &          alf, bta, pi, tau, z0
      common /nonfit/
     &          c2, c1, c0
c
      h   = h0 - .5*q**2
c
c---- set input pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
     &               t,t_r,t_h,t_rr,t_hh,t_rh)
c
      ttc   =  1./(tau*t)
      ttc_t = -1./(tau*t**2)
c
      ph    = phi(ttc)
      phd   = phid(ttc)
c
      s   = alf*log(t) + 2.0*alf*bta*t
     &    - p*pi*(t*phd*ttc_t + ph) - log(p)
c
      return
      end


      subroutine nongamv(h0,r,q, gam,gam_r,gam_q)
c-------------------------------------------------------
c     Returns "equivalent" gamma for BL density profile
c-------------------------------------------------------
      common /nongas/
     &          alf, bta, pi, tau, z0
      common /nonfit/
     &          c2, c1, c0
c
c---- set static enthalpy
      h   = h0 - 0.5*q**2
      h_q =         -q
c
c---- set pressure and temperature and derivatives
      call ngaspt(h,r,p,p_r,p_h,p_rr,p_hh,p_rh,
     &               t,t_r,t_h,t_rr,t_hh,t_rh)
c
c---- set speed of sound squared:  a^2 = dp/dr  (at constant s)
      asq   = p_r  / (1. - p_h/r)
```

```
      asq_r = p_rr / (1. - p_h/r)
     &      - p_r  / (1. - p_h/r)**2 *(p_h/r**2 - p_rh/r)
      asq_h = p_rh / (1. - p_h/r)
     &      + p_r  / (1. - p_h/r)**2 *p_hh/r
c
c
      ttc    =  1./(tau*t)
      ttc_t  = -1./(tau*t**2)
      ttc_tt =  2./(tau*t**3)
c
      ph     = phi(ttc)
      phd    = phid(ttc)
      phdd   = phidd(ttc)
      phddd  = phiddd(ttc)
c
      z    = 1. + p*pi*ph
      z_p =        pi*ph
      z_t =      p*pi*phd*ttc_t
c
      cp    = ( alf*(1.0 + 2.0*bta*t)
     &        + p*pi/tau* phdd*ttc_t ) / z0
      cp_p = (   pi/tau* phdd*ttc_t ) / z0
      cp_t = ( alf*(      2.0*bta  )
     &        + p*pi/tau*(phddd*ttc_t**2 + phdd*ttc_tt) ) / z0
c
      zet    =   h/(cp*t)*(1.0 - p*pi/(t*tau)*phd/z) * z0
      zet_h = 1.0/(cp*t)*(1.0 - p*pi/(t*tau)*phd/z) * z0
      zet_p =   h/(cp*t)*(    -   pi/(t*tau)*phd/z
     &                        - p*pi/(t*tau)*phd/z*(-z_p/z)) * z0
     &      - (zet/cp)*cp_p
      zet_t =   h/(cp*t)*(    - p*pi/(t*tau)*phd/z*(-z_t/z - 1.0/t)
     &                        - p*pi/(t*tau)*phdd*ttc_t/z  ) * z0
     &      - (zet/cp)*cp_t - (zet/t )
c
c
      gam    = asq/(h*zet) + 1.0
      gam_r =                                          asq_r/(h*zet)
      gam_h = asq/(h*zet)*(-zet_h/zet - 1.0/h) + asq_h/(h*zet)
      gam_p = asq/(h*zet)*(-zet_p/zet)
      gam_t = asq/(h*zet)*(-zet_t/zet)
c
      gam_h = gam_p*p_h + gam_t*t_h + gam_h
      gam_r = gam_p*p_r + gam_t*t_r + gam_r
c
      gam_q = gam_h*h_q
c
      return
      end


      subroutine sonic(h0,p0,r0, q,p,r)
c-----------------------------------------------
c     calculates sonic quantities q,p,r
c     from specified sonic quantities h0,p0,r0
```

```
c-----------------------------------------------
      implicit real (m)
      data eps / 1.0e-5 /
c
c---- initialize with perfect gas
      gam = r0*h0 / (r0*h0 - p0)
      gm1 = gam - 1.0
c
      q = sqrt(2.0*h0/(2.0/gm1 + 1.0))
c
      trat = 1.0 + 0.5*gm1
      p = p0*trat**(-gam/gm1)
      r = r0*trat**(-1.0/gm1)
c
c---- converge on non-ideal values by forcing M^2 = 1, and pstag = p0
      do 10 iters=1, 15
        call  nideal(h0,r,q, p  ,p_r  ,p_q,
     &                       msq,msq_r,msq_q )
        call nonstag(h0,r,q, pstag,pstag_r,pstag_q,
     &                       rstag,rstag_r,rstag_q )
        res1 = msq - 1.0
        a11  = msq_r
        a12  = msq_q
c
        res2 = pstag - p0
        a21  = pstag_r
        a22  = pstag_q
c
        detinv = 1.0/(a11*a22 - a12*a21)
        dr = -(res1*a22  - a12 *res2)*detinv
        dq = -(a11 *res2 - res1*a21 )*detinv
c
        dp = p_r*dr + p_q*dq
c
        rlx = 1.0
        if(rlx*dr .gt. 1.5*r) rlx = 1.5*r/dr
        if(rlx*dr .lt. -.6*r) rlx = -.6*r/dr
        if(rlx*dq .gt. 1.5*q) rlx = 1.5*q/dq
        if(rlx*dq .lt. -.6*q) rlx = -.6*q/dq
c
        r = r + rlx*dr
        q = q + rlx*dq
        p = p + rlx*dp
c
        dmax = amax1( abs(dr)/r , abs(dq)/q )
c
        if(dmax .lt. eps) go to 11
c
 10   continue
      write(*,*) 'sonic: convergence failed.  dmax =', dmax
 11   continue
c
      return
      end ! sonic
```

Bibliography

[1] J. H. McMasters, W. H. Roberts, and F. M. Payne. Recent air-freon tests of a transport airplane in high lift configurations. AIAA-88-2034, 1988.

[2] H. W. Liepmann and A. Roshko. *Elements of Gasdynamics*. Wiley, New York, 1957.

[3] B. Wagner and W. Schmidt. Theoretical investigations of real gas effects in cryogenic wind tunnels. *AIAA Journal*, 16(6), June 1978.

[4] M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10), Oct 1987.

[5] W. Anderson. A numerical study on the use of sulfur hexaflouride as a test gas for wind tunnels. AIAA-90-1421, 1990.

[6] O. Coufal. Thermodynamic properties of sulphur hexaflouride in temperature range 298.15 – 30,000 K and pressure range .101325 – 2 MPa. *ACTA Technica CSAV*, 31, 1986.

Appendix B

High-Order Airfoil Farfield Boundary Conditions for Ideal and Non-Ideal Gas Flows

The steady flow around an airfoil away from shock wakes and viscous regions has constant entropy and total enthalpy, and hence is also irrotational. These properties hold whether the fluid is an ideal or a non-ideal gas. The flow can then still be decribed by the velocity potential Φ or the perturbation potential ϕ. Assuming the freestream is aligned with the x-axis, the following relations are obtained.

$$\Phi = q_\infty(x+\phi) \tag{1}$$

$$\nabla\Phi \equiv \vec{q} = q_\infty\left[(1+\phi_x)\hat{\imath} + \phi_y\hat{\jmath}\right] \tag{2}$$

$$q^2 \equiv |\vec{q}|^2 = q_\infty^2\left[(1+\phi_x)^2 + \phi_y^2\right] \tag{3}$$

$$\frac{1}{2}\nabla(q^2) = q\nabla q = q_\infty^2\left[\ (\phi_{xx}+\phi_x\,\phi_{xx}+\phi_y\phi_{xy})\hat{\imath} + (\phi_{xy}+\phi_x\,\phi_{xy}+\phi_y\phi_{yy})\hat{\jmath}\ \right] \tag{4}$$

The governing flow equation is:

$$\nabla\cdot(\rho\nabla\Phi) = 0 \tag{5}$$

$$\text{or} \qquad \nabla^2\Phi = -\frac{\nabla\rho}{\rho}\cdot\nabla\Phi \tag{6}$$

In isentropic flow (s = constant), $\rho = \rho(p)$, so

$$\nabla\rho = \left.\frac{d\rho}{dp}\right|_s \nabla p = -\frac{\rho}{a^2}\,q\nabla q \tag{7}$$

and hence

$$\nabla^2\Phi = \frac{1}{a^2}\,q\nabla q\cdot\nabla\Phi \tag{8}$$

$$a^2\,\nabla^2\phi = q\nabla q\cdot\left[(1+\phi_x)\hat{\imath} + \phi_y\hat{\jmath}\right] \tag{9}$$

where a is the speed of sound. In isentropic, adiabatic flow, the speed of sound is uniquely related to the speed: $a = a(q)$. For a perfect gas, $a(q)$ is given by

$$a^2 = a_\infty^2 - \frac{\gamma-1}{2}\left(q^2 - q_\infty^2\right) \tag{10}$$

while for an imperfect and/or non-ideal gas it is necessary here to linearize $a(q)$ about the freestream conditions.

$$a^2 \simeq a_\infty^2 + \left.\frac{d(a_\infty^2)}{d(q_\infty^2)}\right|_s \left(q^2 - q_\infty^2\right) \tag{11}$$

It is convenient to define an "equivalent" ratio of specific heats γ' for the non-ideal gas as

$$\gamma' \equiv 1 - 2\left.\frac{d(a_\infty^2)}{d(q_\infty^2)}\right|_s \tag{12}$$

so that the $a(q)$ relation for the non-ideal gas can be compactly written as

$$a^2 \simeq a_\infty^2 - \frac{\gamma'-1}{2}\left(q^2 - q_\infty^2\right) \tag{13}$$

For a perfect gas, this reverts to the exact form (10) since in this case $\gamma' = \gamma$. It is interesting to note that γ' can easily be less than unity for heavy gases such as sulfur hexafluoride, while invariably $\gamma > 1$ for perfect gases.

Substituting for a^2, q^2, and $q\nabla q$ in equation (9), we obtain

$$\left[a_\infty^2 - \frac{\gamma'-1}{2}q_\infty^2\left(2\phi_x + \phi_x^2 + \phi_y^2\right)\right][\phi_{xx} + \phi_{yy}] = q_\infty^2\left[(1+\phi_x)(\phi_{xx} + \phi_x\,\phi_{xx} + \phi_y\phi_{xy}) + \phi_y(\phi_{xy} + \phi_x\,\phi_{xy} + \phi_y\phi_{yy})\right] \tag{14}$$

$$\left[1 - \frac{\gamma'-1}{2}M_\infty^2\,2\phi_x\right][\phi_{xx} + \phi_{yy}] = M_\infty^2\left[\phi_{xx} + 2\phi_x\,\phi_{xx} + 2\phi_y\phi_{xy}\right] + \mathcal{O}\left(\phi^3\right) \tag{15}$$

$$(1 - M_\infty^2)\phi_{xx} + \phi_{yy} = M_\infty^2\left[(\gamma'+1)\phi_x\phi_{xx} + (\gamma'-1)\phi_x\phi_{yy} + 2\phi_y\phi_{xy}\right] + \mathcal{O}\left(\phi^3\right) \tag{16}$$

where $M_\infty = q_\infty/a_\infty$ is the freestream Mach number.

Equation (16) is the 2D second-order Prandtl-Glauert equation which governs small-perturbation non-ideal compressible potential flows. It has the same form as the equation for a perfect ideal gas as derived in references [4] and [5], except that the usual ratio of specific heats γ is replaced by the "equivalent" value γ' defined by equation (12). Wagner and Schmidt [1] have considered the first-order version of equation (16) using γ' in lieu of γ.

In terms of the Prandtl-Glauert coordinates

$$\bar{x} = x/\beta \tag{17}$$

$$\bar{y} = y \tag{18}$$

$$\bar{r}^2 = \bar{x}^2 + \bar{y}^2 \tag{19}$$

$$\theta = \arctan\frac{\bar{y}}{\bar{x}} \tag{20}$$

where $\beta = \sqrt{1 - M_\infty^2}$, the general solution to equation (16) is

$$\phi = \frac{-\Gamma}{2\pi}\theta + \frac{\Sigma}{2\pi}\ln\bar{r} + \frac{D_x}{2\pi}\frac{\cos\theta}{\bar{r}} + \frac{D_y}{2\pi}\frac{\sin\theta}{\bar{r}} + \left(\frac{\Gamma M_\infty}{2\pi}\right)^2\left[k_1\frac{\ln\bar{r}}{\bar{r}} - k_2\frac{\cos 3\theta}{\bar{r}}\right] \tag{21}$$

where

$$k_1 = \frac{1}{4}\left(\frac{\gamma'+1}{\beta^3} + \frac{3-\gamma'}{\beta}\right) \qquad k_2 = \frac{1}{16}\left(\frac{\gamma'+1}{\beta^3} + \frac{\gamma'-1}{\beta}\right). \tag{22}$$

Terms of order $1/\bar{r}^2$ and above have been discarded.

In a flow solver, the circulation Γ can be determined either directly from the lift per unit span L' (Euler or Navier-Stokes code),

$$\Gamma = \frac{L'}{\rho_\infty q_\infty^2} \tag{23}$$

or indirectly by specifying a Kutta condition (potential solver or MSES). The source strength Σ can be determined from the total profile drag per unit span D', or from the asymptotic mass defect behind the airfoil including the shock wake.

$$\Sigma = \frac{D'}{\rho_\infty q_\infty^2} \tag{24}$$

In the case of a potential solver, D' should not include the wave drag since there is no shock wake (unless an entropy correction scheme is employed). Note that Γ and Σ here have units of length since ϕ in (1) corresponds to a unit freestream speed.

Cole and Cook [5] give explicit expressions for the doublet coefficients D_x and D_y in terms of field integrals over the domain. Unfortunately, these expressions are unwieldy and for a non-ideal gas would be rather expensive. A simpler and economical approach is to iteratively update D_x and D_y by minimizing the mismatch between $\nabla\Phi$ and the velocity $\vec{q}_{\text{solution}}$ from the flow solver on the outer boundary. The approach taken in reference [4], for example, is to minimize the integral

$$I = \frac{1}{2}\int |\nabla\Phi \times \vec{q}_{\text{solution}}|^2 \, dx \tag{25}$$

taken over the outermost streamlines. The doublet terms in the farfield expansion (21) decay faster than the others, and so can be neglected for sufficiently distant outer boundaries. However, retaining them greatly reduces the sensitivity of the solution to domain size, especially for transonic flows [6].

With its term coefficients defined, equation (21) gives a very accurate representation of the perturbation potential ϕ away from the airfoil. The gradient of equation (21) accurately gives the total velocity $\vec{q}$ via relation (2). Either ϕ, $\vec{q}$, or an appropriate derived quantity may then be imposed at the outer domain boundary as a high-order boundary condition. A potential solver would typically impose ϕ or $\partial\phi/\partial n$, whereas an Euler or Navier-Stokes solver would typically impose the flow angle at the inflow and pressure at the outflow, both being determined from $\nabla\Phi$.

Appendix C

Shape Parameter Relations for Ideal and Non-Ideal Gas Flows

The major influence of compressibility on boundary layer behavior is a non-uniform density profile, which alters the layer's response to pressure gradients. In an integral scheme for adiabatic flows, this effect is mostly captured by the correlation between shape parameter H, the kinematic shape parameter H_k, and the edge Mach number M_e. The shape parameters are defined as

$$H \equiv \frac{\int(1-RU)\,dy}{\int(1-U)RU\,dy} \qquad H_k \equiv \frac{\int(1-U)\,dy}{\int(1-U)U\,dy} \tag{26}$$

where $U(y)$ and $R(y)$ are the velocity and density profiles.

$$U = \frac{u}{u_e} \qquad R = \frac{\rho}{\rho_e} \tag{27}$$

Since the velocity profile $U(y)$ and hence H_k are only weakly affected by compressibility, reduction of the density profile $R(y)$ near the wall due to adiabatic heating will increase H as can be seen from its definition (26). In turn, the von-Karman integral momentum equation

$$\frac{d\theta}{dx} = \frac{C_f}{2} - (2 + H - M_e^2)\frac{\theta}{u_e}\frac{du_e}{dx} \tag{28}$$

shows that an increase in H will increase the momentum thickness growth rate $d\theta/dx$ for a given adverse pressure gradient. The integral boundary layer formulation in MSES (and its precursor ISES [4]) employs a correlation of the form $H_k(H, M_e)$ for air. This is re-derived for the non-ideal gas model as follows.

As developed in Appendix A, the state equation of a non-ideal gas can be written as

$$\frac{p}{\rho \mathcal{R} T} = Z(p,T) \tag{29}$$

while the corresponding caloric equation in differential form is

$$dh = \bar{c}_p(T)\,dT + d[p\mathcal{F}(T)] \simeq \bar{c}_p(T)\,dT + p\mathcal{F}'(T)\,dT = c_p(p,T)\,dT \tag{30}$$

where the approximation is made on the basis that $dp \simeq 0$ across a boundary layer. Linearizing the caloric equation across the boundary layer we have

$$h - h_e = c_p(T - T_e) \tag{31}$$

$$\frac{T_e}{T} - 1 = \left(\frac{h_e}{h} - 1\right)\frac{h}{c_p T} \simeq \left(\frac{h_e}{h} - 1\right)\frac{h_e}{c_{p_e} T_e} \tag{32}$$

The non-ideality factor Z for most non-ideal gases has the form

$$Z(p,T) = 1 + \frac{p}{p_c}\,\phi(T_c/T) \tag{33}$$

where p_c and T_c are the critical pressure and temperature. This can likewise be linearized about the edge conditions as follows.

$$Z - Z_e = \frac{p_e}{p_c}\phi'_e\left(\frac{T_c}{T} - \frac{T_c}{T_e}\right) \tag{34}$$

$$\frac{Z_e}{Z} = 1 - \frac{p_e}{p_c}\frac{T_c}{T_e}\left(\frac{T_e}{T} - 1\right)\frac{\phi'_e}{Z} \tag{35}$$

Combining this with equation (32), we have

$$\frac{Z_e}{Z} = 1 - \frac{p_e}{p_c}\frac{T_c}{T_e}\left(\frac{h_e}{h} - 1\right)\frac{h_e}{c_{p_e}T_e}\frac{\phi'_e}{Z_e}. \tag{36}$$

Using the equation of state (29), the density profile is then related to the T and Z profiles as

$$\begin{aligned} R = \frac{\rho}{\rho_e} &= \frac{T_e}{T}\frac{Z_e}{Z}\frac{p}{p_e} \\ &= \frac{T_e}{T}\frac{Z_e}{Z} \end{aligned} \tag{37}$$

with the usual boundary layer approximation $p \simeq p_e$ being made. Using relations (32) and (36), the density profile can be written in terms of the enthalpy profile alone.

$$R = \left[1 + \left(\frac{h_e}{h} - 1\right)\frac{h_e}{c_{p_e}T_e}\right]\left[1 - \frac{p_e}{p_c}\frac{T_c}{T_e}\left(\frac{h_e}{h} - 1\right)\frac{h_e}{c_{p_e}T_e}\frac{\phi'_e}{Z_e}\right] \tag{38}$$

$$= 1 + \left(\frac{h_e}{h} - 1\right)\frac{h_e}{c_{p_e}T_e}\left(1 - \frac{p_e}{p_c}\frac{T_c}{T_e}\frac{\phi'_e}{Z_e}\right) + \mathcal{O}\left[(h_e/h - 1)^2\right] \tag{39}$$

$$R \simeq 1 + \left(\frac{h_e}{h} - 1\right)\zeta \tag{40}$$

where

$$\zeta = \frac{h_e}{c_{p_e}T_e}\left(1 - \frac{p_e}{p_c}\frac{T_c}{T_e}\frac{\phi'_e}{Z_e}\right) \tag{41}$$

For turbulent adiabatic boundary layer flows, it is reasonable to assume a constant stagnation enthalpy across the layer, although this is strictly true only for a turbulent Prandtl number of unity. Since the turbulent diffusion mechanisms of momentum and heat are essentially the same in a gas (convection by eddies), turbulent Prandtl numbers are typically close to unity. Hence, the assumption of constant stagnation enthalpy is reasonable. With h_0 denoting the stagnation enthalpy, the velocity and static enthalpy profiles are then related by

$$\frac{h_e}{h} = \frac{h_0 - u_e^2/2}{h_0 - u^2/2} = \frac{1 - \frac{u_e^2}{2h_0}}{1 - \frac{u_e^2}{2h_0}U^2} \tag{42}$$

$$\frac{h_e}{h} - 1 = \frac{\frac{u_e^2}{2h_0}(U^2 - 1)}{1 - \frac{u_e^2}{2h_0}U^2} \tag{43}$$

and the density and velocity profiles are then related by

$$R = 1 + \frac{\frac{u_e^2}{2h_0}}{1 - \frac{u_e^2}{2h_0}U^2}(U^2 - 1)\zeta. \tag{44}$$

Since u_e^2/h_0 and ζ are both functions of the edge Mach number M_e, the density profile (44) implicitly defines H_k in terms of H and M_e. To obtain this relation in closed form, it is necessary to assume a small-defect profile

$$U = 1 - \epsilon \quad ; \quad \epsilon \ll 1 \tag{45}$$

so that the density profile can be approximated by

$$R = 1 + \frac{\frac{u_e^2}{2h_0}}{1 - \frac{u_e^2}{2h_0}} (-2\epsilon)\,\zeta + \mathcal{O}(\epsilon^2) \tag{46}$$

$$R = 1 - (\gamma_v - 1) M_e^2\, \epsilon\, \zeta \tag{47}$$

with the convenient "viscous" equivalent ratio of specific heats γ_v *defined* by

$$\frac{\gamma_v - 1}{2} M_e^2\, \zeta = \frac{\frac{u_e^2}{2h_0}}{1 - \frac{u_e^2}{2h_0}} \tag{48}$$

$$\gamma_v \equiv 1 + \frac{a_e^2}{h_0 - u_e^2/2} \frac{1}{\zeta} \tag{49}$$

with a_e being the speed of sound at the boundary layer edge.

The shape parameter H now becomes

$$\begin{aligned} H &= \frac{\int \{1 - [1 - (\gamma_v - 1) M_e^2\, \epsilon\,](1 - \epsilon)\}\, dy}{\int \{\epsilon [1 - (\gamma_v - 1) M_e^2\, \epsilon\,](1 - \epsilon)\}\, dy} \\ &= \frac{\int \epsilon\, dy + (\gamma_v - 1) M_e^2 \int \epsilon (1 - \epsilon)\, dy}{\int \epsilon (1 - \epsilon)\, dy - (\gamma_v - 1) M_e^2 \int \epsilon^2 (1 - \epsilon)\, dy} \\ &= \frac{\int \epsilon\, dy}{\int \epsilon (1 - \epsilon)\, dy} + (\gamma_v - 1) M_e^2 + \mathcal{O}(\epsilon^2) \\ &\simeq H_k + (\gamma_v - 1) M_e^2 \end{aligned} \tag{50}$$

The required shape parameter correlation is therefore

$$H_k = H - (\gamma_v - 1) M_e^2. \tag{51}$$

In the limiting case of a perfect gas, $\gamma_v = \gamma$. For $\gamma = 1.4$ (air), MSES presently uses Whitfield's correlation [7] in this case is

$$H_k = \frac{H - 0.29 M_e^2}{1 + 0.113 M_e^2} \tag{52}$$

$$= H - 0.4 M_e^2 + \mathcal{O}(M_e^4) \tag{53}$$

which is seen to be consistent with the more general non-ideal gas result (51). Whitfield's particular form (52), however, is reportedly more accurate for Prandtl numbers somewhat less than unity

where the total enthalpy profile is not quite uniform as was assumed here. It is therefore appropriate to put correlation (51) into Whitfield's form, while also incorporating the Prandtl number. The final shape parameter correlation is

$$H_k(H, M_e) = \frac{H - Pr(\gamma_v - 1)M_e^2}{1 + (1 - Pr)(\gamma_v - 1)M_e^2} \tag{54}$$

which reduces to Whitfield's form for $\gamma_v = 1.4$, $\zeta = 1$, and $Pr = 0.7$, and to the non-ideal gas form (51) for $Pr = 1$ which was assumed in its derivation.

It noteworthy that for most heavy gases $\gamma_v - 1$ is considerably smaller than for air. For SF_6 with stagnation conditions at STP and $M_e = 1$, for example, $\gamma_v - 1 = 0.17$ for SF_6 and $\gamma_v - 1 = \gamma - 1 = 0.4$ for air. Hence, the influence of M_e in SF_6 is smaller, and H values near a shock in SF_6 will be smaller than those in air. The smaller H values in turn reduce the boundary layer's response to adverse pressure gradients as discussed above. The airfoil will therefore be more resistant to Mach drag-divergence in SF_6 than in air.

For simplicity, the implementation of the shape parameter correlation (54) in MSES assumes that γ_v is constant, being evaluated from (49) at sonic edge conditions: $a_e = u_e = a^*$, $M_e = 1$. Given the degree of approximation used in deriving (54), it is felt that neglecting the already weak dependence of γ_v on u_e is justified. Freezing γ_v at the sonic conditions is judged appropriate since its effect on H becomes significant only for M_e close to unity.