AE 5830 Project: Numerical Solutions to the Inviscid Supersonic Wedge and Supersonic Cone Problems

Alexander Fangman
May 13, 2019

Abstract

The flow over an inviscid supersonic wedge at zero angle of attack and an inviscid supersonic cone at zero angle of attack were solved using six different numerical techniques discussed in class. These techniques include: root finding, numerical differentiation, solving sets of non-linear equations, solving sets of linear equations, interpolation, and solving ordinary differential equations (ODE's). A finite difference approach was used to discretize and solve the governing equations for the wedge case, and numerical integration of ODE's was used to solve the governing equation for the cone case. Methods derived and results obtained were verified using analytic expressions and existing data sets.

Contents

Nomenclature

$(\epsilon_n)_r$	Relative successive error at the n^{th} iteration
$\bar{F}_{i,j}$	Vector quantity denoting values at the (i,j) mesh point
β	Oblique shockwave angle
γ	Ratio of specific heats
μ	Mach wave angle
ρ	Density
θ	Flow deflection angle
M	Mach number
M_n	Normal component of Mach number
p	Pressure
p_0	Total (stagnation) pressure
R	Ideal gas constant
T	Temperature
T_0	Total (stagnation) temperature
u	x-component of velocity
v	y-component of velocity, Prandtl-Meyer angle
V'_θ	Nondimensional transverse component of velocity
V'_r	Nondimensional radial component of velocity

Introduction

Although the modeling and analysis of the zero angle of attack supersonic wedge and supersonic cone cases are relatively simple, they provide many important results, which can be applied and built upon in more complex cases. The wedge and cone problems were chosen because they directly correlate to my OURE project. Additionally, all code was written in C++ to refresh my memory in its usage, as it is the primary language utilized by SU2.

Before continuing, a brief outline of the paper will now be given. This paper is split into two main sections: the solution of the wedge problem and the solution of the cone problem. These sections are subsequently organized as follows: a brief description of the problem will be given, followed by the formulation of the numerical problem and how numerical techniques were employed. Each problem will then conclude with a results and discussion section, which will detail important findings and remark on any issues faced during the process.

Inviscid Supersonic Wedge at Zero Angle of Attack

Description of Problem

The flow over a supersonic 2D wedge is characterized by an attached oblique shock wave, which causes a discontinuity in the flow field variables. The freestream streamlines are deflected such that they become parallel to the wedge surface directly behind the shock. For a CPG (intermolecular forces are neglected and specific heats are assumed to be constant), the solution becomes dependent solely on the freestream Mach number, the deflection angle, and the type of fluid (i.e. air for this case). The derivation is not shown, but analytic expressions for the ratio of the thermodynamic quantities before and after the shock can be written as

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)M_{1n}^2}{2+(\gamma-1)M_{1n}^2} \tag{1}$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1}\big(M_{1n}^2 - 1\big) \tag{2}$$

$$\frac{T_2}{T_1} = \left[1 + \frac{2\gamma}{\gamma+1}\big(M_{1n}^2 - 1\big)\right]\left[\frac{2+(\gamma-1)M_{1n}^2}{(\gamma+1)M_{1n}^2}\right] \tag{3}$$

where the subscript '1' denotes the freestream value, and the subscript '2' denotes the shock region value. For this problem, the following values were chosen: $M_1 = M_\infty = 2.0$, $\gamma = 1.4$, and $\theta = 5°$. Eq.'s (1), (2), and (3) were used to verify numerical results.

Formulation of Numerical Problem

Since the inviscid flow assumption is used, the problem is governed by the Euler equations. The 2D, inviscid, steady, zero body-force, zero source term Euler equations can be written in conservation form as

$$\frac{\partial \bar{F}}{\partial x} + \frac{\partial \bar{G}}{\partial y} = 0 \tag{4}$$

where the flux vectors are

$$\bar{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \dfrac{\gamma}{\gamma - 1} p u + \rho u \dfrac{u^2 + v^2}{2} \end{bmatrix} \quad \text{and} \quad \bar{G} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ p v \dfrac{\gamma}{\gamma - 1} + \rho v \dfrac{u^2 + v^2}{2} \end{bmatrix}$$

Using the flux variables as dependent variables make the conservation form more suitable for shock-capturing. The strong conservation form was necessary so that all x-derivative terms could be grouped to one side; this allowed for the formulation of the downstream marching scheme. The four unknowns are density, x-velocity, y-velocity, and pressure. The temperature was obtained using the equation of state

$$T = \frac{p}{\rho R} \tag{5}$$

Because the finite difference approach used requires a uniform grid, the physical domain was mapped onto a uniform computational domain. A sketch of this transformation is shown in Fig. 1.

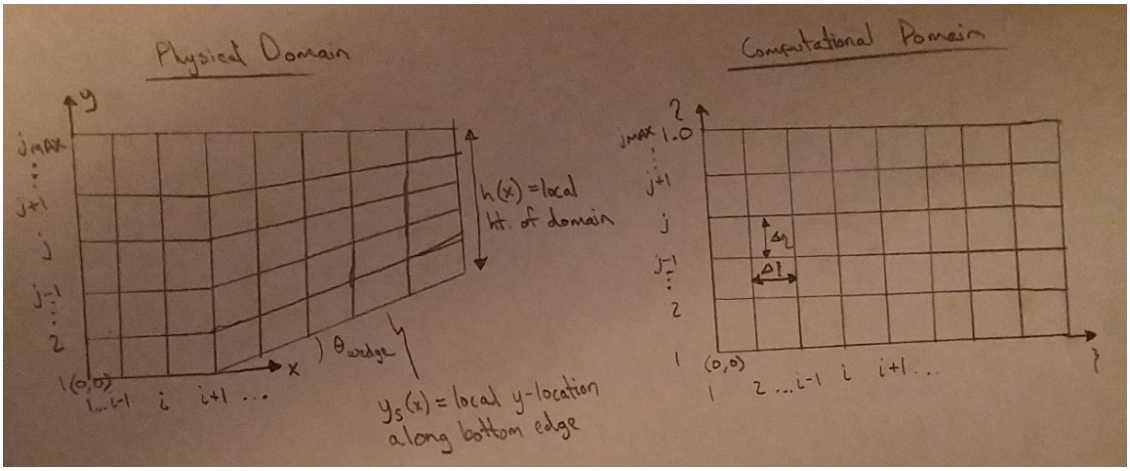

Figure 1: Physical and computational domains.

The boundary-fitted coordinate system used to define the transformation can be written as

$$\xi = x \tag{6}$$

$$\eta = \frac{y - y_s(x)}{h(x)} \tag{7}$$

where $h(x)$ is the local physical domain height and $y_s(x)$ is the local y-location of the lower boundary.

For the sake of brevity, the derivation of the transformation metrics and transformation of the governing equations will not be shown. Refer to Ch. 8 of Anderson [1] for a detailed explanation of these methods (remember that the $(\partial\eta/\partial x)$ term will not be the same as the expansion corner example). With this in mind, Eq. (4) can be rewritten in terms of the computational domain as

$$\frac{\partial \bar{F}}{\partial \xi} = -\left[\left(\frac{\partial \eta}{\partial x}\right)\frac{\partial \bar{F}}{\partial \eta} + \frac{1}{h}\frac{\partial \bar{G}}{\partial \eta}\right] \tag{8}$$

where

$$\frac{\partial \eta}{\partial x} = \begin{cases} 0 & 0 \leq x < x_{ramp} \\ (\eta - 1)\dfrac{tan(\theta)}{h(x)} & x_{ramp} \leq x \leq x_{max} \end{cases}$$

MacCormack's Method was used to solve the system by marching in the x-direction from an initial data line (inlet). This approach is an appropriate technique since the flow is hyperbolic in nature. MacCormack's Method is a second-order accurate predictor-corrector finite difference scheme. The main idea of MacCormack's Method is to calculate the next step's variables using an average derivative

$$\phi_{i+1,j} = \phi_{i,j} + \left(\frac{\partial \phi}{\partial x}\right)_{avg} \Delta x$$

Here, ϕ represents an arbitrary scalar variable such as temperature. $(\partial\phi/\partial x)_{avg}$ is determined as the mean of the derivative approximated by first-order forward differences (predictor step) and the derivative approximated by first-order backward differences using predicted values (corrector step). For this problem, the forward and backward differences correspond to spatial differences in the y-direction. Note that the use of this two-step method retains second-order accuracy while circumventing the calculation of any second derivative terms.

Boundary conditions were required to close the solution. Since the upper boundary is outside the influence of the shockwave, values were kept constant at these nodes. For the nodes along the lower boundary, the flow was forced to be tangential to the wall. This was done by "rotating" the flow through a local Prandtl-Meyer expansion to adjust its direction (if needed). After rotating the flow, corrected thermodynamic values were calculated from isentropic relations. First-order one-sided differences were used at the lower boundary in the formulation of the MacCormack's Method.

To ensure stability of the solution, an appropriate step size, $\Delta\xi$, had to be chosen. The next grid point must be within the zone of influence of the current point. Using the concept of characteristics, this qualitative statement can be written as a stability criterion formula. Applied to the current problem, Anderson [1] states that, "the characteristic lines are Mach lines in the flow, which are at the Mach angle μ relative to the streamline direction". This is shown in Fig. 2, which is taken directly from Ch. 3 of Anderson.

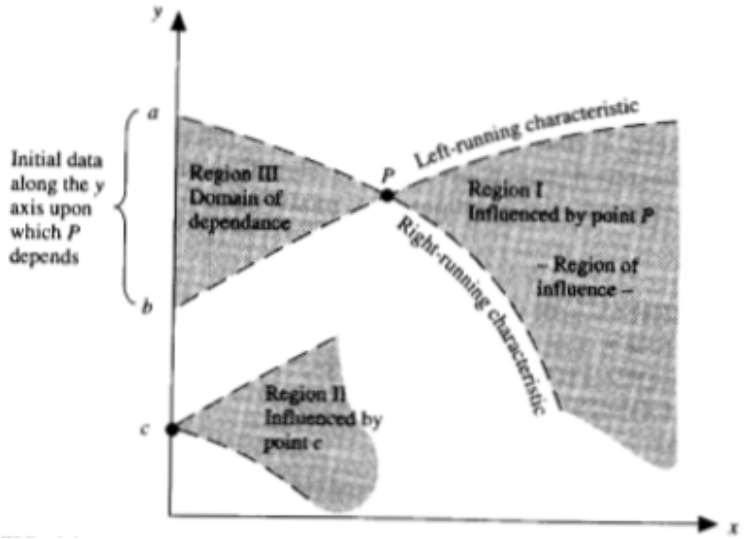

Figure 2: Domain and boundaries for the solution of hyperbolic equations. Two-dimensional steady flow.

It can be shown that for stability, the following criterion must be met

$$\Delta x = \Delta \xi = C \frac{\Delta y}{|tan(\theta \pm \mu)|_{max}} \tag{9}$$

where the Courant number, $C \leq 1$.

Recall from Eq. (8) that the wedge causes a discontinuity in the solution. This can lead to potentially detrimental oscillations in the solution if left unchecked. To counteract this issue, an artificial viscosity term was added to the predictor and corrector steps of the MacCormack's Method. Following Anderson's example, the artificial viscosity term was defined as

$$(\overline{SF})_{i,j} = \frac{C_y |p_{i,j+1} - 2p_{i,j} + p_{i,j-1}|}{p_{i,j+1} + 2p_{i,j} + p_{i,j-1}} \left[\bar{F}_{i,j+1} - 2\bar{F}_{i,j} + \bar{F}_{i,j-1}\right] \tag{10}$$

where C_y was a chosen constant used to vary the viscosity's influence. Since the viscosity term is composed of two second-order accurate central differences, the viscosity term is $O(\Delta\eta^4)$. Although artificial viscosity tends to smear results as a trade-off for an increase in stability, the loss of accuracy was minimal in this case. As will be shown later, the final results (aside from wall values) were within 1% percent of analytic values: an appropriate error level for this case.

An abbreviated outline of how the solution was obtained will now be given. Refer to the appropriate section of the attached code to see exactly how each step was implemented.

Outline of Solution Method

Set up geometry and initial conditions:

1. Input mesh size and freestream conditions, including: pressure, temperature, Mach number, Courant number, coefficient of viscosity, and number of nodes in the y-direction.

2. Calculate $\Delta\eta$ and $\Delta\xi$.
3. Adjust $\Delta\xi$ so that the solution stops marching at the end of the domain (ensuring to round down in step size if needed).
4. Calculate initial flux vectors, $\bar{F}$ and $\bar{G}$. Note that each node contains their own $\bar{F}$ and $\bar{G}$.

   Begin loop to march downstream:
5. Calculate the local height, $h(\xi)$.
6. Calculate the $(\partial\eta/\partial x)$ metric for each node at the current ξ-location.
7. Calculate current $\bar{G}$ vectors.
8. Predictor step:
   - Calculate $(\partial\bar{F}/\partial\xi)$ using forward differences.
   - Calculate predictor artificial viscosity terms, $(\overline{SF})_p$.
   - Calculate predicted $\bar{F}$ vectors: $\bar{F}_{p_{i,j}} = (\bar{F})_{i,j} + \left(\frac{\partial\bar{F}}{\partial\xi}\right)_{i,j}\Delta\xi + (\overline{SF})_{p_{i,j}}$
   - Decode the primitive variables and use them to calculate predicted $\bar{G}$ vectors. (The more appropriate way to do this would be to directly use the components of $\bar{F}$ to determine the components of $\bar{G}$ since it would bypass any error accrued from the iterative solution of the non-linear equations. I did not implement this to save time; I had already developed a function to decode the primitive variables so that they could be output to the solution file).
9. Corrector step:
   - Calculate $(\partial\bar{F}/\partial\xi)$ using backward differences and the predicted values obtained in step 8.
   - Calculate corrector artificial viscosity terms, $(\overline{SF})_c$.
   - Calculate average derivatives: $\left(\frac{\partial\bar{F}}{\partial\xi}\right)_{avg} = \frac{1}{2}\left[\left(\frac{\partial\bar{F}}{\partial\xi}\right)_p + \left(\frac{\partial\bar{F}}{\partial\xi}\right)_c\right]$
10. Advance the interior nodes to the next ξ-location: $\bar{F}_{i+1,j} = (\bar{F})_{i,j} + \left(\frac{\partial\bar{F}}{\partial\xi}\right)_{avg_{i,j}}\Delta\xi + (\overline{SF})_{c_{i,j}}$ for $j = 2, 3, ..., j_{max} - 1$
11. Apply Euler wall and constant boundary conditions.
12. Write the variables of interest to the solution file.
13. Repeat again beginning at step 5 until the entire domain has been solved for.

Numerical Methods Used

Five different numerical methods discussed in class were used in the solution of this problem: numerical differentiation, solving sets of non-linear equations, solving sets of linear equations, interpolation, and root finding.

A quick aside: the version I uploaded to GitHub as part of my OURE project does not contain the code for solving non-linear sets of equations, solving linear sets of equations, or the interpolation methods. As will be discussed in subsequent sections, these were not required for the formulation of this problem since more efficient algebraic methods could be developed through manipulation

of the equations and/or using known analytic expressions.

A brief description of how each numerical method was implemented will now be given.

Numerical Differentiation

The governing equations given in Eq. (8) were discretized using a first-order forward difference in the predictor step. After multiplying the negative through, the discretized equations become

$$\left(\frac{\partial \bar{F}}{\partial \xi}\right)_{p_{i,j}} = \left(\frac{\partial \eta}{\partial x}\right)\frac{(\bar{F})_{i,j} - \bar{F})_{i,j+1}}{\Delta\eta} + \frac{1}{h}\frac{(\bar{G})_{i,j} - (\bar{G})_{i,j+1}}{\Delta\eta} \tag{11}$$

A first-order backward difference was used to discretize the governing equations in the corrector step. After multiplying the negative through, the discretized equations in the corrector step become

$$\left(\frac{\partial \bar{F}}{\partial \xi}\right)_{c_{i,j}} = \left(\frac{\partial \eta}{\partial x}\right)\frac{(\bar{F})_{p_{i,j-1}} - \bar{F})_{p_{i,j}}}{\Delta\eta} + \frac{1}{h}\frac{(\bar{G})_{p_{i,j-1}} - (\bar{G})_{p_{i,j}}}{\Delta\eta} \tag{12}$$

where the predicted values are obtained using the derivative in Eq. (11) to linearly extrapolate. Remember that these are vector equations and that they represent the four scalar equations: continuity, x-momentum, y-momentum, and energy.

Solving Sets of Non-linear Equations

To decode the primitive variables from the flux variables, Newton's Method for the solution of a non-linear set of equations was used. Recall that

$$\bar{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \dfrac{\gamma}{\gamma - 1} p u + \rho u \dfrac{u^2 + v^2}{2} \end{bmatrix}$$

and that after completing a step in the MacCormack's Method, values for $F_1,\ F_2,\ F_3$, and F_4 are known. To solve this system for the solution vector, $\bar{x} = \{\rho,\ u,\ v,\ p\}^T$, the given system was first written in the form

$$\bar{f}(\bar{x}) = \begin{bmatrix} f_1(\bar{x}) \\ f_2(\bar{x}) \\ f_3(\bar{x}) \\ f_4(\bar{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where

$$\begin{aligned} f_1(\bar{x}) &= \rho u - F_1 \\ f_2(\bar{x}) &= \rho u^2 + p - F_2 \\ f_3(\bar{x}) &= \rho u v - F_3 \\ f_4(\bar{x}) &= \frac{\gamma}{\gamma - 1} p u + \rho u \frac{u^2 + v^2}{2} - F_4 \end{aligned}$$

The system was then linearized and written as

$$-\bar{f}^k = J^k \Delta \bar{x}^k \tag{13}$$

where the Jacobian is

$$J(\bar{x}) = \begin{bmatrix} u & \rho & 0 & 0 \\ u^2 & 2\rho u & 0 & 1 \\ uv & \rho v & \rho u & 0 \\ u\dfrac{u^2+v^2}{2} & \dfrac{\rho}{2}(3u^2+v^2) + \dfrac{\gamma}{\gamma-1}p & \rho uv & \dfrac{\gamma}{\gamma-1}u \end{bmatrix}$$

$\Delta \boldsymbol{x}^k$ was determined via the linear solver. The updated solution vector was then calculated as $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k$. This process was repeated until the convergence criterion

$$\frac{||\bar{f}^k||_2}{||\bar{f}^0||_2} \leq 10^{-16} \quad \text{or} \quad iterations \geq N_{MAX}$$

was satisfied, where $||\bar{f}^k||_2$ represents the L2-norm of the function vector at the k^{th} step. N_{MAX} was set as 100.

The initial solution vector, $\bar{x}^0$, was constructed using the variable values from the previous ξ-location. This was done to minimize the possibility of converging to an incorrect solution.

As was mentioned previously, the use of Newton's Method here is not required since algebraic expressions for the solution vector variables can be obtained. The density can be calculated as

$$\rho = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

where

$$A = \frac{F_3^2}{2F_1} - F_4$$
$$B = \frac{\gamma}{\gamma - 1} F_1 F_2$$
$$C = -\frac{\gamma + 1}{2(\gamma - 1)} F_1^3$$

The other variables can then be determined using

$$u = \frac{F_1}{\rho}$$
$$v = \frac{F_3}{F_1}$$
$$p = F_2 - F_1 u$$

The addition of Newton's Method greatly increased the runtime. This is because the four equation, 4 unknown system had to solved at every mesh point (i.e. a linear system had to be solved "$Number\_of\_Mesh\_Points * Number\_of\_Newton\_Method\_Iteration$" times). This is why it is left off of the version I added to GitHub.

Solving Sets of linear Equations

Gaussian Elimination with partial pivoting was used to solve the linear system in Eq. (13) for $\Delta\bar{x}^k$. A direct method was chosen because the size of the Jacobian is small, and it is not diagonally dominant.

Interpolation

A natural cubic spline was used to approximate the Prandtl-Meyer angle as a function of the Mach number using data obtained from the NACA TR1135 [3]. Seventeen equally spaced points from Mach 1.0 to Mach 5.0 (i.e. $h = 0.25$) were used to define the approximation. To solve the tri-diagonal system for the 'c' coefficients of the local cubic polynomials, the same Gaussian Elimination with partial pivoting code was used.

To ensure an appropriate approximation was being used, values were compared to the analytic formulation for the Prandtl-Meyer function

$$v(M) = \sqrt{\frac{\gamma+1}{\gamma-1}} tan^{-1}\left[\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)}\right] - tan^{-1}\left(\sqrt{M^2-1}\right) \tag{14}$$

Root Finding

The Secant Method was used to calculate the Mach number from the Prandtl-Meyer function approximation. This was used to calculate a corrected Mach number after rotating the flow at the boundary nodes through an appropriate Prandtl-Meyer angle.

Writing this statement as a root funding problem yields

$$S(M) - v = 0 \tag{15}$$

where v is the known Prandtl-Meyer angle after adjusting the flow direction and $S(M)$ refers to the cubic spline approximation of Eq. (14).

The two initial guesses for the Mach number were the calculated Mach number before correcting and this same value plus 0.5 (it was added to ensure both initial guesses were still within the interval [1.0, 5.0]). The convergence criterion specified for the Secant Method was

$$(\epsilon_n)_r = \frac{|M_{n+1} - M_n|}{|M_n|} < 10^{-6} \quad \text{or} \quad iterations \geq N_{MAX}$$

where N_{MAX} was set to 100.

Results and Discussion

Post-processing was done using the "Post_process.m" MATLAB script. For the case that is shown here, the following values were used (freestream conditions correspond to 100,000 feet): $p_\infty = 1090.16\,Pa$, $T_\infty = 227.13\,K$, $\rho_\infty = 0.0167207\,kg/m^3$, $M_\infty = 2$, $C = 0.5$, $\theta_{wedge} = 5°$, $C_y = 0.8$, and 101 nodes in the y-direction. Fig.'s (3), (4), and (5) depict contour plots of various variables of interest: Mach number, pressure, and temperature respectively.

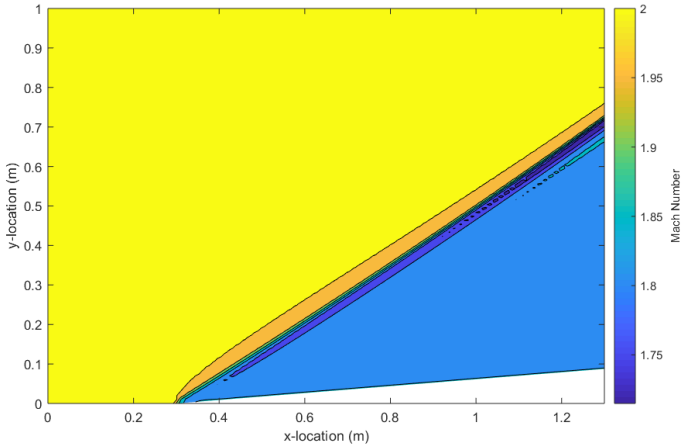

Figure 3: Mach number contour, 5 degree wedge at Mach 2.

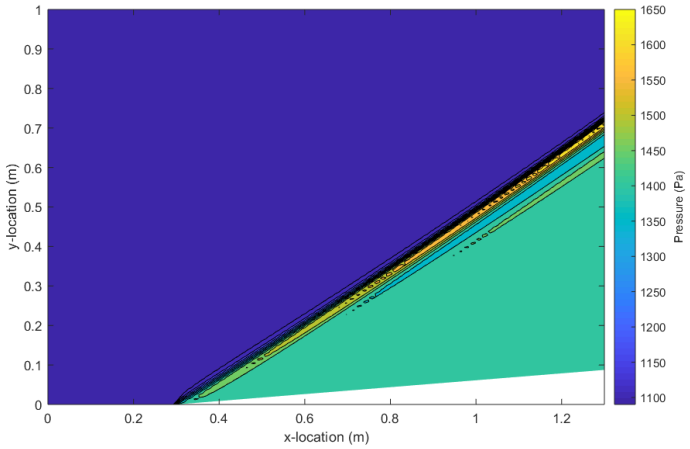

Figure 4: Pressure contour, 5 degree wedge at Mach 2.

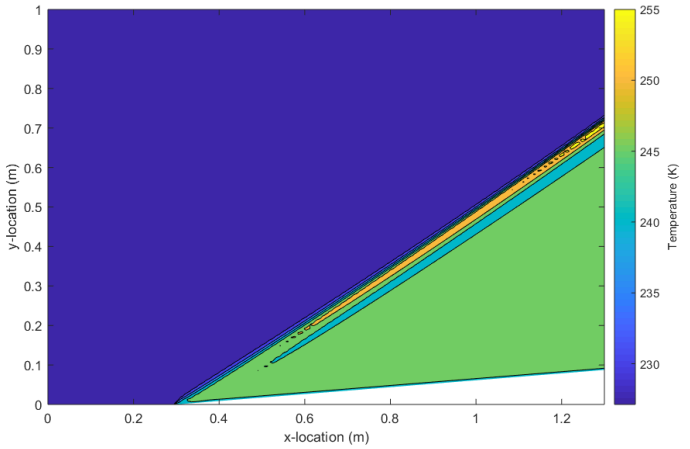

Figure 5: Temperature contour, 5 degree wedge at Mach 2.

A few noteworthy points can be made by examining the contour plots. It is clear that the shockwave is stretched over multiple grid points. Although not an issue here, this may lead to erroneous shock region values if the number of y-direction grid points and/or the Courant number are chosen such that the oscillations resulting from the discontinuity are allowed to propagate downstream. Another important observation is that error values at the wall nodes are slightly larger than at surrounding nodes. This is a result of the combination of using a first-order one-sided difference at the wall and using local Prandtl-Meyer expansions as a way to "correct" these values. The Courant number and number of nodes in the y-direction was specifically chosen to ensure the resulting error was minimized (less than 2% error at wall nodes for this case). Table 1 compares the average shock region results to analytic values obtained from Eq.'s (1), (2), and (3).

Table 1: Wedge Results

Variable	Code Result	Analytic Result	Percent Error
Pressure (Pa)	1434	1434.004030	0.0003
Temperature (k)	245.8	245.783114	0.0069
Density (kg/m^3)	0.02032	0.020325301	0.0261
Mach number	1.821	1.821254	0.0139

It can be seen that the shock region values are extremely close to the analytic values: a direct result of the relatively fine grid used ($\approx$ 15,000 grid points). A more efficient way to achieve accurate results at the wall boundary without decreasing the mesh spacing would have been to use a higher-order one-sided difference for these points and/or use a different, more robust approach to enforce the Euler wall boundary condition.

One important convergence issue encountered was the influence of large oscillations created by the discontinuity. I had originally planned to simulate a 20° wedge; however, the oscillations for this case were large enough such that its impact on the flux variables caused the resulting nonlinear set of equations to have no real solution. This is why I settled on a 5° wedge for this example.

Time constraints restricted the amount of testing and error analysis I was able to do with this and other cases; however, I was able to accurately simulate small wedge angles up to approximately Mach 4.5 using appropriate mesh and viscosity inputs. Larger Mach numbers were not simulated because, barring certain freestream conditions, hypersonic phenomena begin to occur (i.e. results are not valid), and the Prandtl-Meyer interpolation created was only valid for Mach [1, 5].

Inviscid Supersonic Cone at Zero Angle of Attack

Description of Problem

Like the 2D wedge, the inviscid supersonic cone is characterized by an attached oblique shockwave. However, due to 3D effects, the shockwave over the cone is much weaker (i.e. the shock angle is smaller). Streamlines are initially deflected by the shock, then continuously curve as they asymptotically become parallel with the cone surface.

The Taylor-Macoll solution [4] was used. It assumes a finite cone can be modeled as a semi-infinite cone. This is done to make the argument that flow field variable values are constant along rays which emanate from the cone vertex. Axisymmetric flow is also assumed.

Refer to Ch. 13 of Anderson [2] to see a detailed derivation: the nondimensional governing equation for the inviscid axisymmetric supersonic cone is

$$\frac{\gamma-1}{2}\left[1-V_r'^2-\left(\frac{dV_r'}{d\theta}\right)^2\right]\left[2V_r'+\frac{dV_r'}{d\theta}cot\theta+\frac{d^2V_r'}{d\theta^2}\right]-\frac{dV_r'}{d\theta}\left[V_r'\frac{dV_r'}{d\theta}+\frac{dV_r'}{d\theta}\left(\frac{d^2V_r'}{d\theta^2}\right)\right]=0 \quad (16)$$

where the nondimensional transverse component of velocity is

$$V_\theta'=\frac{dV_r'}{d\theta} \quad (17)$$

Eq. (16) is a second-order ordinary differential equation with the nondimensional radial velocity, V_r', as the dependent variable. The nondimensional velocity can be calculated with Eq. (18).

$$V'=\left[\frac{2}{(\gamma-1)M^2}+1\right]^{-1/2} \quad (18)$$

Note that the nondimensional velocity can also be calculated from its radial and transverse components as

$$V'=\sqrt{V_r'^2+V_\theta'^2} \quad (19)$$

Formulation of Numerical Problem

To use the numerical integration techniques discussed in class, the second-order ODE in Eq. (16) was first broken into two coupled first-order ODE's. Letting

$$x_1\equiv V_r' \quad (20)$$

$$x_2\equiv\frac{dV_r'}{d\theta} \quad (21)$$

The first ODE is then

$$x_1'=x_2 \quad (22)$$

The second ODE is acquired by substituting Eq.'s (20) and (21) into Eq. (16) and algebraically solving for x_2'. For the sake of brevity, the derivation will not be shown; the final expression was determined to be

$$x_2' = \frac{cot(\theta)a(x_2 - x_1^2 x_2 - x_2^3) - x_1 x_2^2(2a+1) + 2ax_1(1-2x_1^2)}{a(x_1^2 + x_2^2 - 1) + x_2^2} \tag{23}$$

where the constant $a = (\gamma - 1)/2$ was defined to make the equation more compact.

To begin the integration process, velocity components directly behind the oblique shock wave are used as initial conditions: $x_1(0) = V'_{r_{a.s.}}$ and $x_2(0) = V'_{\theta_{a.s.}}$.

An abbreviated outline of how the solution was obtained will now be given. Refer to the appropriate section of the attached code to see exactly how each step was implemented.

Outline of Solution Method

Set up geometry and initial conditions:

1. Prompt user for freestream conditions and cone half-vertex angle.
2. Determine two initial guesses for the oblique shock wave angle to begin the Secant Method: these were chosen as the cone half-vertex angle plus two degrees and the corresponding 2D wedge oblique shock angle minus five degrees.

   Begin root finding loop:
3. Calculate the deflection angle and Mach number behind the shock using oblique shock wave analysis:

   $$\delta = tan^{-1}\left(2cot\beta\left[\frac{M_1^2 sin^2\beta - 1}{M_1^2(\gamma + cos(2\beta)) + 2}\right]\right) \quad \text{and} \quad M_2 = \frac{M_{2n}}{sin(\beta - \delta)}$$

   where

   $$M_{2n} = \sqrt{\frac{1 + \left[(\gamma-1)/2\right]M_{1n}^2}{\gamma M_{1n}^2 - \left[(\gamma-1)/2\right]}}$$
4. Calculate the nondimensional velocity behind the shock from Eq. (18).
5. Determine components of the nondimensional velocity from geometry: $V_r' = V'cos(\beta - \delta)$ and $V_\theta' = -V'sin(\beta - \delta)$. Note that these are the initial values for the integration process.
6. Numerically integrate the ODE's in the interval [θ_{cone}, β] using the four-stage Runge-Kutta method. Note that the resulting values correspond to V_r' and V_θ' for the solutions to Eq. (22) and Eq. (23) respectively.
7. Determine V_θ at the surface. For the surface to be impermeable, it should zero at this point.
8. Update the shock angle using the Secant Method and repeat again beginning at step 3 until the convergence criterion is satisfied.

   Solve for the rest of flow field:

9. Use Eq. (19) to determine the nondimensional velocity magnitude along each ray.
10. Calculate the Mach number along each ray from Eq. (18).
11. Calculate the rest of the flow field variables using isentropic relations:
    - The total temperature is constant because of the shock is assumed to be adiabatic. The freestream total temperature and total pressure are

$$T_0 = T_1\left[1 + \frac{\gamma-1}{2}M_1^2\right] \quad \text{and} \quad p_{0_{b.s.}} = p_1\left(1 + \frac{\gamma-1}{2}M_1^2\right)^{\gamma/(\gamma-1)}$$

    - The total pressure after the shock can be related to the freestream total pressure with

$$p_{0_{a.s.}} = p_{0_{b.s.}}\left[\frac{(\gamma+1)M_{1n}^2}{(\gamma-1)M_{1n}^2+2}\right]^{\gamma/(\gamma-1)}\left[\frac{\gamma+1}{2\gamma M_{1n}^2-(\gamma-1)}\right]^{1/(\gamma-1)}$$

    - Isentropic relations are used to determine pressure and temperature along rays:

$$T = T_0\left[1 + \frac{\gamma-1}{2}M^2\right]^{-1} \quad \text{and} \quad p = p_{0_{a.s.}}\left(1 + \frac{\gamma-1}{2}M_1^2\right)^{-\gamma/(\gamma-1)}$$

    - The equation of state is used to calculate density:

$$\rho = \frac{p}{RT}$$

Numerical Methods Used

Two different numerical methods discussed in class were used in the solution of this problem: root finding and numerical integration of ODE's. A brief description of how each was implemented will be given.

Root Finding

The Secant Method was used to iteratively solve for an oblique shock angle which ensured the transverse component of velocity, V_θ, was zero at the surface of the cone. This is equivalent to solving $f(\beta) = 0$, where $f(\beta)$ is the transverse velocity at the cone's surface. Note that there is no analytic expression for $f(\beta)$.

The convergence criterion was specified as

$$|f(\beta_n)| < 10^{-5} \quad \text{or} \quad iterations \geq N_{MAX}$$

where N_{max} was set to 100.

Solution of ODE's

To numerically integrate Eq.'s (22) and (23), the four-stage Runge-Kutta Method was used. Since the equations are coupled, they had to be solved simultaneously. This simply means that the Runge-Kutta Method (shown here only for x_1)

$$x_{1_{i+1}} = x_{1_i} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\Delta\theta$$

where

$$k_1 = f(\theta_i, x_{1_i}, x_{2_i})$$
$$k_2 = f(\theta_i + \frac{1}{2}\Delta\theta, x_{1_i} + \frac{1}{2}k_{1,1}\Delta\theta, x_{2_i} + \frac{1}{2}k_{1,2}\Delta\theta)$$
$$k_3 = f(\theta_i + \frac{1}{2}\Delta\theta, x_{1_i} + \frac{1}{2}k_{2,1}\Delta\theta, x_{2_i} + \frac{1}{2}k_{2,2}\Delta\theta)$$
$$k_4 = f(\theta_i + \Delta\theta, x_{1_i} + k_{3,1}\Delta\theta, x_{2_i} + k_{3,2}\Delta\theta)$$

had to be applied to both equations before moving to the next step: $\theta = \theta + \Delta\theta$. The second subscript on the 'k' values (slopes) denote which ODE's 'k' value was used. The step size was chosen as $\Delta\theta = -0.001^\circ$ (remember that the interval of integration is [θ_{cone}, β] and that initial values are given at β). Depending on the shock and cone angle, this could require a relatively large amount of function evaluations; however, with this step size, the code still runs within a few seconds and error values are within an appropriate level.

Results and Discussion

Although the entire shock region is solved for during the solution process, only the shock angle and quantities on the cone surface will be analyzed and discussed. Table 2 shows the code results compared to the results given by Virginia Tech's compressible flow calculator [5] (this was much quicker than reading the TR 1135 plots). Since there is no closed form expression for the shock angle, and Virginia Tech's web-page does not state what criteria is set when solving, or if they use data tables, these percent difference values must be taken with a grain of salt. Similar to the wedge problem, freestream conditions correspond to an altitude of 100,000 feet.

Table 2: Cone Results

Case	Variable	Code Result	Flow Calc.	Percent Diff.
$M_\infty = 2, \theta_{cone} = 20^\circ$	Shock angle (deg.)	37.7965	37.8266	0.079605
	Surface Pressure (Pa)	2083.94	2086.8000	0.137146
	Surface Mach number	1.56772	1.56672	0.063807
$M_\infty = 5, \theta_{cone} = 20^\circ$	Shock angle (deg.)	24.9431	24.9785	0.141822
	Surface Pressure (Pa)	6059.80	6075.37	0.256610
	Surface Mach number	3.37513	3.37201	0.092484
$M_\infty = 4, \theta_{cone} = 10^\circ$	Shock angle (deg.)	17.7151	17.7491	0.191743
	Surface Pressure (Pa)	2059.69	2067.46	0.376531
	Surface Mach number	3.53052	3.52768	0.080474

Again, it is difficult to draw set conclusions from this data; however, it would be expected that the surface quantities should have the largest error since they are calculated using the final integration values. A fourth-order method with a small step size was used to mitigate this error. On the other hand, the round-off error may have become prevalent due to this small step size. I did not take the time to observe the solution with a varying step size.

There were multiple issues encountered while working on this problem, some of which I did not resolve fully:

1. The Runge-Kutta step size needed to be sufficiently small to allow the Secant Method's convergence criterion to be possible. This is depicted graphically in Fig. (6) and Fig. (7) for an arbitrary case.

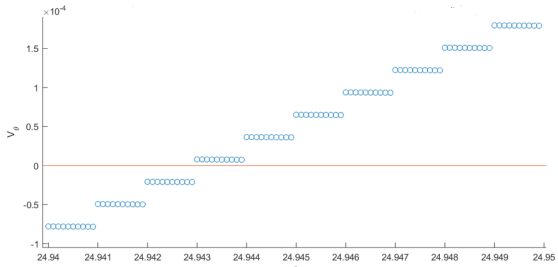

Figure 6: $f(\beta)$ before decreasing step size.

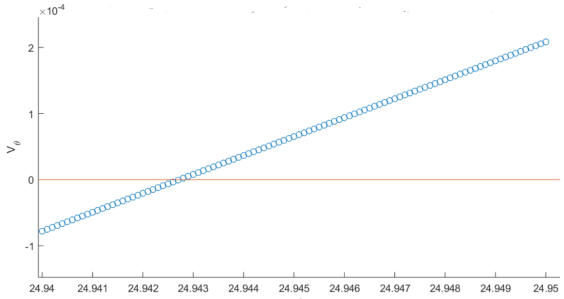

Figure 7: $f(\beta)$ after decreasing step size.

Before I decreased $\Delta\theta$, the Secant Method would not converge because the function value jumped back and forth between the values in Fig. (6).

2. The second initial guess for the Secant Method is the 2D wedge oblique shock angle minus five degrees. Plotting $f(\beta)$ shows that for large shock angle flows, $f(\beta)$ becomes sporadic slightly before the 2D wedge angle. Fig. (8) shows an example of this: $f(\beta)$ is plotted for a cone with a semi-vertex angle of $20°$ in Mach 2 flow. The corresponding 2D wedge oblique shock angle is $53.4°$. If one of these "random" values is calculated as the other initial guess, it causes the Secant Method to break down and give incorrect results. I did not give myself ample time to change this. Both initial guesses should be fixed if the code is to be used in the future.

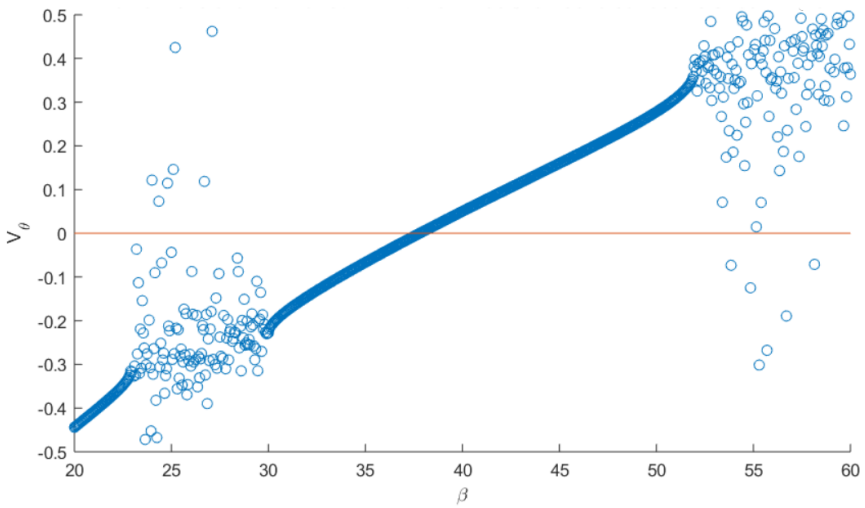

Figure 8: Example of $f(\beta)$ "blowing up" before 2D wedge shock angle.

3. The code does not check for a detached shockwave. It will simply return an error and exit. I did not spend time fixing this.

Conclusions

As stated previously, both of these solution methods have their own drawbacks. Aside from numerical errors, the wedge code is only able to simulate small angles, and the cone code does not detect detached shocks. These codes are, however, very accurate for the cases in which they are valid. Tables (1) and (2) attest to this fact.

In conclusion, I enjoyed this project... but I definitely let out a sigh of relief when the wedge code finally worked.

References

[1] Anderson, John D., Jr. Computational fluid dynamics: the basics with applications. *Mechanical Engineering Series. McGraw-HILL*, 1995.

[2] Anderson, John D., Jr. *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.

[3] Ames Research Lab. Equations, tables, and charts for compressible flow. *NACA Technical Report 1135*, 1953.

[4] G.I. Taylor and J.W. Maccoll. The air pressure on a cone moving at high speed. *Proc. Roy. Soc.*, 1933.

[5] Virginia Tech. Compressible aerodynamics calculator, http://www.dept.aoe.vt.edu/ devenpor/aoe3114/calc.html.