**Conical Flow: Supersonic Flow Past Un-Yawed Cone Computer Project**

**Class: ARO 3111 – Gas Dynamics & Highspeed Aerodynamics**

**Section #02**

**Instructor: Dr. Tony Lin**



**Justin T. Millsap**

Aerospace Engineering Department

California Polytechnic University, Pomona

April 11th, 2024

**Table of Contents**

[**List of Symbols:** 1](#_Toc163559706)

[**Summary of Key Equation** 2](#_Toc163559707)

[**Executive Summary:** 3](#_Toc163559708)

[**Plots** 4](#_Toc163559709)

[**Table of Computed Data** 7](#_Toc163559710)

[**Appendix A:** Hand Calculations of flow properties at station 2 and station C. 9](#_Toc163559711)

[**Appendix B**: MATLAB Code for Taylor-Maccoll & Runge Kutta 10](#_Toc163559712)

# **List of Symbols:**

M = Mach Number

= Normal Component of Mach Number

P = Pressure

Density

T = Temperature

= Specific Heat Ratio (1.4)

= Shock Angle

= Deflection Angle

= Cone Angle

= Freestream

o = Stagnation values

c = Surface of the cone

a = Speed of Sound (1089 ft/s)

V = Flow Velocity

= Normal Velocity Component

= Radial Velocity Component

A diagram of a triangle with arrows

Description automatically generated

# **Summary of Key Equation**

A close up of a math problem

Description automatically generated with medium confidence**Taylor-Maccoll Equation**:

**A math equations on a white background

Description automatically generatedInitial Conditions at Station 2:**

**Runge-Kutta Equations:**

**Flow Property Equations:**

A mathematical equation with numbers and symbols

Description automatically generated, A mathematical equation with numbers

Description automatically generated, A mathematical equation with numbers

Description automatically generated

# **Executive Summary:**

The objective of this computer project is to write a computer program that calculates the supersonic flow properties over an un-yawed cone. The solution includes the usage of the Taylor-Maccoll equation and the utilization of the 4th – order Runge-Kutta method to solve for specific parameters.

This computer program uses initial conditions of Free Stream Mach (M∞), Shockwave Angle (θs), and the gas constant (γ). It then follows by solving for the deflection angle using the θ-β-M relation. Once these parameters have been found, the initial conditions Vr & Vθ can be found using the geometry of V∞. These initial conditions are then used in the 4th – order Runge-Kutta numerical solver to solve for the radial and normal components of flow velocity with respect to a desired iteration of shockwave angles. These values are then used to calculate the Mach Numbers (M) that correspond to different shockwave angels. Lastly, the Mach numbers are used to calculate the flow properties in the conical flow region using planar shock relations.

The angle of the shock cone was unknown, but it was apparent that once the normal component (Vθ) of the flow velocity (V∞), reaches a non-negative value, this means that the deflection angle is equivalent to the cone angle. In this specific case, the cone angle was found to be 7°. The plots of the flow properties (pressure, density, and temperature ratios) across the cone, showed that when θs < 12° , the properties were constant at a value of 1. Once the shock wave passed 12°, each ratio increased at different values, but in increased the same fashion. The Mach number was constant up to 12°, then rapidly dropped once 12° had passed. Lastly, Vθ showed a linear increase throughout the change in shockwave angles.

These results from the Taylor-Maccoll computer program turned out to be accurate as it matched hand calculations of the same algorithm. After an oblique shock, the pressure, density, and temperature all increase so this satisfies the results from the computer program. As the shockwave angle decreases this will result in the Mach number decreasing.

In conclusion, the integration of the Taylor-Maccoll equations with the 4th-order Runge-Kutta solver presents a robust and reliable method for computing the flow properties over a cone. This approach solves for the normal and radial components of the flow velocity, which are essential for deriving the overall flow velocity. These calculated velocities lay the ground for calculations, offering a precise and dependable solution to understand and predict the behavior of supersonic flows over conical geometries.

# **Plots**

**A graph with a line going up

Description automatically generated**

**Figure 1:** Vθ  vs. θs

**A graph with a line

Description automatically generated**

**Figure 2:**  Mach Number vs. θs

**A graph with lines and numbers

Description automatically generated**

**Figure 3:**  P/P1 , ρ/ ρ1  , & T/T1  vs. θ

# **Table of Computed Data**

**Table 1:** Vr , Vθ  , Mach Number, and flow properties for various Shock Wave (θₛ) angles

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| θₛ | Vr | Vθ | Mach # | P∕P1 | ρ∕ρ1 | T∕T1 |
| 20 | 0.87083 | -0.317 | 6 | 1 | 1 | 1 |
| 19 | 0.87623 | -0.3017 | 6 | 1 | 1 | 1 |
| 18 | 0.88136 | -0.2864 | 6 | 1 | 1 | 1 |
| 17 | 0.88623 | -0.2709 | 6 | 1 | 1 | 1 |
| 16 | 0.89082 | -0.2554 | 6 | 1 | 1 | 1 |
| 15 | 0.89514 | -0.2399 | 6 | 1 | 1 | 1 |
| 14 | 0.89919 | -0.2242 | 6 | 1 | 1 | 1 |
| 13 | 0.90297 | -0.2085 | 6 | 1 | 1 | 1 |
| 12.1 | 0.90613 | -0.1943 | 6 | 1 | 1 | 1 |
| 12 | 0.90647 | -0.1927 | 5.50631 | 1 | 1 | 1 |
| 12 | 0.91657 | -0.1368 | 5.49848 | 1.66413 | 1.43356 | 1.16083 |
| 11.9 | 0.9168 | -0.1338 | 5.49115 | 1.67841 | 1.44234 | 1.16367 |
| 11.8 | 0.91703 | -0.1309 | 5.48426 | 1.6919 | 1.45061 | 1.16634 |
| 11.7 | 0.91726 | -0.1281 | 5.47773 | 1.70472 | 1.45845 | 1.16885 |
| 11.6 | 0.91748 | -0.1252 | 5.47153 | 1.71694 | 1.46591 | 1.17124 |
| 11.5 | 0.9177 | -0.1225 | 5.46561 | 1.72865 | 1.47305 | 1.17352 |
| 11.4 | 0.91791 | -0.1197 | 5.45995 | 1.7399 | 1.47989 | 1.1757 |
| 11.3 | 0.91811 | -0.117 | 5.45453 | 1.75074 | 1.48647 | 1.17779 |
| 11.2 | 0.91832 | -0.1143 | 5.44931 | 1.76121 | 1.49281 | 1.17979 |
| 11.1 | 0.91851 | -0.1116 | 5.44429 | 1.77133 | 1.49894 | 1.18173 |
| 11 | 0.91871 | -0.1089 | 5.43945 | 1.78114 | 1.50486 | 1.18359 |
| 10.9 | 0.91889 | -0.1063 | 5.43477 | 1.79066 | 1.5106 | 1.1854 |
| 10.8 | 0.91908 | -0.1036 | 5.43025 | 1.7999 | 1.51616 | 1.18714 |
| 10.7 | 0.91926 | -0.101 | 5.42588 | 1.80888 | 1.52157 | 1.18883 |
| 10.6 | 0.91943 | -0.0983 | 5.42165 | 1.81763 | 1.52681 | 1.19047 |
| 10.5 | 0.9196 | -0.0957 | 5.41755 | 1.82613 | 1.53192 | 1.19206 |
| 10.4 | 0.91976 | -0.0931 | 5.41357 | 1.83442 | 1.53688 | 1.1936 |
| 10.3 | 0.91992 | -0.0904 | 5.40972 | 1.84249 | 1.5417 | 1.1951 |
| 10.2 | 0.92008 | -0.0878 | 5.40598 | 1.85036 | 1.5464 | 1.19656 |
| 10.1 | 0.92023 | -0.0852 | 5.40235 | 1.85802 | 1.55097 | 1.19797 |
| 10 | 0.92038 | -0.0825 | 5.39883 | 1.86549 | 1.55542 | 1.19934 |
| 9.9 | 0.92052 | -0.0799 | 5.39542 | 1.87276 | 1.55975 | 1.20068 |
| θₛ | Vr | Vθ | Mach # | P∕P1 | ρ∕ρ1 | T∕T1 |
| 9.8 | 0.92066 | -0.0773 | 5.39212 | 1.87984 | 1.56396 | 1.20197 |
| 9.7 | 0.92079 | -0.0746 | 5.38891 | 1.88674 | 1.56806 | 1.20323 |
| 9.6 | 0.92092 | -0.072 | 5.38581 | 1.89345 | 1.57204 | 1.20445 |
| 9.5 | 0.92104 | -0.0693 | 5.38281 | 1.89997 | 1.5759 | 1.20564 |
| 9.4 | 0.92116 | -0.0666 | 5.37991 | 1.9063 | 1.57965 | 1.20678 |
| 9.3 | 0.92127 | -0.064 | 5.37711 | 1.91244 | 1.58329 | 1.20789 |
| 9.2 | 0.92138 | -0.0613 | 5.37441 | 1.9184 | 1.58681 | 1.20896 |
| 9.1 | 0.92149 | -0.0586 | 5.37181 | 1.92415 | 1.59021 | 1.21 |
| 9 | 0.92159 | -0.0559 | 5.36931 | 1.92972 | 1.59349 | 1.211 |
| 8.9 | 0.92168 | -0.0531 | 5.36692 | 1.93508 | 1.59665 | 1.21196 |
| 8.8 | 0.92177 | -0.0504 | 5.36463 | 1.94023 | 1.59969 | 1.21288 |
| 8.7 | 0.92186 | -0.0476 | 5.36245 | 1.94517 | 1.6026 | 1.21376 |
| 8.6 | 0.92194 | -0.0449 | 5.36038 | 1.94989 | 1.60537 | 1.2146 |
| 8.5 | 0.92201 | -0.0421 | 5.35842 | 1.95439 | 1.60802 | 1.2154 |
| 8.4 | 0.92208 | -0.0393 | 5.35658 | 1.95864 | 1.61052 | 1.21616 |
| 8.3 | 0.92215 | -0.0364 | 5.35487 | 1.96266 | 1.61287 | 1.21687 |
| 8.2 | 0.92221 | -0.0336 | 5.35327 | 1.96642 | 1.61508 | 1.21754 |
| 8.1 | 0.92227 | -0.0307 | 5.35181 | 1.96991 | 1.61713 | 1.21815 |
| 8 | 0.92232 | -0.0278 | 5.35049 | 1.97312 | 1.61901 | 1.21872 |
| 7.9 | 0.92236 | -0.0249 | 5.3493 | 1.97603 | 1.62072 | 1.21923 |
| 7.8 | 0.92241 | -0.022 | 5.34827 | 1.97863 | 1.62224 | 1.21969 |
| 7.7 | 0.92244 | -0.019 | 5.3483 | 1.98091 | 1.62357 | 1.22009 |
| 7.6 | 0.92247 | -0.016 | 5.3474 | 1.98283 | 1.6247 | 1.22043 |
| 7.5 | 0.9225 | -0.0129 | 5.3467 | 1.98439 | 1.62561 | 1.2207 |
| 7.4 | 0.92252 | -0.0099 | 5.3462 | 1.98555 | 1.62629 | 1.22091 |
| 7.3 | 0.92253 | -0.0067 | 5.3458 | 1.98629 | 1.62672 | 1.22104 |
| 7.2 | 0.92254 | -0.0036 | 5.3457 | 1.98659 | 1.6269 | 1.22109 |
| 7.1 | 0.92254 | -0.0004 | 5.3458 | 1.9864 | 1.62679 | 1.22106 |
| 7 | 0.92254 | 0.00284 | 5.3457 | 1.9866 | 1.6269 | 1.22109 |

# **Appendix A:** Hand Calculations of flow properties at station 2 and station C.

A white background with text

Description automatically generated

# **Appendix B**: MATLAB Code for Taylor-Maccoll & Runge Kutta

% ARO - 3111 Gas Dynamics & Highspeed Aerodynamics

% Computer Assignment - Taylor-Maccoll Program

% Written By: Justin Millsap

% Date: 03/21/2024

% Tool Version: R2023b

% other .m files required: RungeKutta.m

% Other files required: NONE

% Reference: John D. Anderson - Modern Compressible Flow

tic

%% Nomenclature

%{

- Station 1 is the free stream

- Station 2 is right behind the shock

- Station c is at the surface of the cone

- Subscript o is stagnation values

- Cone angles is theta\_c

- Conical shock angle is theta\_s or beta

- Initial turning of the streamline as the flow crosses the shock from 1 to 2

is delta which is < theta\_c

%}

clc;

clear

%% Givens

theta\_s = 12:-0.1:0; % Shock wave angle [deg]

theta\_s = deg2rad(theta\_s);

M\_1 = 6;

gamma = 1.4;

beta = 12; % [deg]

beta = deg2rad(beta);

% Solve for M2 & Delta (flow deflection angle)

M\_n1 = M\_1 \* sin(beta);

% Theta-Beta-M relation [Eq 4.17]

num\_tbm = (M\_1)^2 \* (sin(beta))^2 - 1;

den\_tbm = (M\_1)^2 \* (gamma + cos(2\*beta)) + 2;

delta = atan(2 \* cot(beta) \* (num\_tbm / den\_tbm));

% Solve for M\_n2

num\_Mn2 = (M\_n1)^2 + (2 / (gamma - 1));

den\_Mn2 = ((2 \* gamma) / (gamma - 1)) \* (M\_n1)^2 - 1;

M\_n2 = sqrt(num\_Mn2 / den\_Mn2);

% Solve for M2

M\_2 = M\_n2 / sin(beta - delta);

% Solve for V\_prime, V\_r\_prime, V\_theta\_prime

V\_prime = (2 / ((gamma - 1) \* (M\_2)^2) + 1)^(-0.5);

V\_theta\_prime = sin(beta - delta) \* V\_prime\*(-1);

V\_r\_prime = cos(beta - delta) \* V\_prime;

%%

[V\_r , V\_theta] = RungeKutta(V\_r\_prime, V\_theta\_prime , theta\_s, gamma);

V\_r = V\_r(1:51);

V\_theta = V\_theta(1:51);

V = sqrt(V\_r.^2 + V\_theta.^2);

M = zeros(1,160);

for i = 1:51

M(i+108) = sqrt(2/((V(i)^-2 - 1)\*(gamma-1)) );

end

M(1:109) = M\_1;

M(160) = 5.3457;

angles\_for\_plots\_Mach = 22.9:-0.1:7;

angles\_for\_plots\_Mach(109) = 12;

%% Ratios

% Pressure ratio (Isolating for P/P1)

P2overP1 = (1 + ((2\*gamma)/(gamma+1))\*(M\_n1^2 -1) ) ; % Equation 3.30

PoverPo2 = (1+((gamma-1)/2).\*M.^2).^(-gamma/(gamma-1)); % Equation 3.30

P2overPo2 = (1+((gamma-1)/2).\*M\_2.^2).^(-gamma/(gamma-1)); % Equation 3.30

PoverP1 = (PoverPo2/P2overPo2)\*P2overP1;

% Density ratio (Isolating for rho/rho1)

rho2overrho1 = (((gamma+1)\* M\_n1^2) / ((gamma-1) \* (M\_n1^2)+2)) ;

rhooverrhoo2 = (1 +((gamma-1)/2).\*M.^2).^(-1/(gamma-1)); % Equation 3.30

rho2overrhoo2 = (1 +((gamma-1)/2).\*M\_2^2).^(-1/(gamma-1)); % Equation 3.30

rhooverrho1 = (rhooverrhoo2/rho2overrhoo2)\*rho2overrho1;

% Tempertuare ratio (Isolating for T/T1)

T = 1 + (gamma-1)/2.\*M.^2;

T1 = 1+ (gamma -1)\*M\_1^2 / 2;

ToverT1 = T1./T;

figure(1);

set(gcf, 'Position', [100, 100, 800, 600]);

hold on;

plot(angles\_for\_plots\_Mach, M(1:160), 'LineWidth', 2);

set(gca, 'XDir', 'reverse');

ylim([4.8 6.2]);

yticks(4.8:0.2:6.2);

xlim([7 20]);

xticks(7:1:20);

ylabel('Mach Number', 'FontSize', 14);

xlabel('\theta\_s', 'FontSize', 14);

title('Mach Number vs \theta\_s', 'FontSize', 16);

set(gca, 'FontSize', 12);

grid on;

hold off;

angles\_for\_plots\_ratio = 18:-0.1:7;

rhooverrho1 = rhooverrho1(50:160);

PoverP1 = PoverP1(50:160);

ToverT1 = ToverT1(50:160);

PoverP1(1:60) = 1;

rhooverrho1(1:60) = 1;

ToverT1(1:60) = 1;

angles\_for\_plots\_ratio(60) = 12;

figure(2);

set(gcf, 'Position', [100, 100, 800, 600]);

hold on;

plot(angles\_for\_plots\_ratio, PoverP1, 'Color', 'b', 'LineWidth', 2);

plot(angles\_for\_plots\_ratio, rhooverrho1, 'Color', '[1, 0.5, 0]', 'LineWidth', 2);

plot(angles\_for\_plots\_ratio, ToverT1, 'Color', 'r', 'LineWidth', 2);

legend('P/P\_1', '\rho/\rho\_1', 'T/T\_o', 'Location', 'best', 'FontSize', 12);

set(gca, 'XDir', 'reverse');

ylim([0 3]);

yticks(0:0.2:3);

xlim([7 20]);

xticks(7:1:20);

ylabel('P/P1 , \rho/\rho1 , T/T1', 'FontSize', 14);

xlabel('\theta\_s [Degrees]', 'FontSize', 14);

title('P/P1 , \rho/\rho1 , T/T1 vs \theta\_s', 'FontSize', 16);

set(gca, 'FontSize', 12);

grid on;

hold off;

angles\_for\_plots\_Vtheta = [20 , 19 , 18 , 17 , 16 , 15 , 14 , 13 , 12.1 , 12]\*pi/180;

Vtheta = -V\_prime\*sin(angles\_for\_plots\_Vtheta);

Vr = V\_prime\*cos(angles\_for\_plots\_Vtheta);

angles\_for\_plots\_Vtheta = [angles\_for\_plots\_Vtheta theta\_s(1:51)]\*180/pi;

V\_theta = [Vtheta V\_theta];

figure(3);

hold on;

set(gcf, 'Position', [100, 100, 800, 600]);

plot(angles\_for\_plots\_Vtheta, V\_theta, 'LineWidth', 2);

set(gca, 'XDir', 'reverse');

legend('V\_\theta', 'FontSize', 12);

ylim([-0.5 0.1]);

yticks([-0.5:0.1:0.2]);

xlim([7 20]);

xticks(7:1:20);

ylabel('V\_\theta', 'FontSize', 14);

xlabel('\theta\_s [Degrees]', 'FontSize', 14);

title("V\_\theta vs. \theta\_s", 'FontSize', 16);

set(gca, 'FontSize', 12);

grid on;

hold off;

V\_r = [Vr V\_r];

M = M(110:160);

MachCalc = [6 6 6 6 6 6 6 6 6 6];

Mach = [MachCalc M];

ones = [1 1 1 1 1 1 1 1 1 1];

T\_overT1 = [ones ToverT1(61:111)];

rho\_over\_rho1 = [ones rhooverrho1(61:111)];

P\_overP1 = [ones PoverP1(61:111)];

toc

fprintf('θ\_s | V\_r | V\_θ | Mach # | P/P\_1 | ρ/ρ\_1 | T/T\_1\n');

fprintf('-------------------------------------------------------------------------\n');

for i = 1:61

fprintf('%7.1f | %5.3f | %7.3f | %6.2f | %5.3f | %9.3f | %5.3f\n', ...

angles\_for\_plots\_Vtheta(i), V\_r(i), V\_theta(i), Mach(i), P\_overP1(i), rho\_over\_rho1(i), T\_overT1(i));

end

%% Runge Kutta Function

function [V\_r, V\_theta] = RungeKutta(V\_r\_prime, V\_theta\_prime, theta\_s, gamma)

%{

RungeKutta method that solves for V\_r & V\_theta.

The inputs are the initial conditions of V\_r and V\_theta (V\_r\_prime, V\_theta\_prime),

a vector of theta values (theta\_s), and the specific heat ratio (gamma).

%}

h = -0.1 \* pi / 180; % Convert step size to radians if theta\_s is in degrees

V\_r = zeros(1, length(theta\_s)); % Initialize V\_r

V\_theta = zeros(1, length(theta\_s)); % Initialize V\_theta

% Set the initial conditions

V\_r(1) = V\_r\_prime;

V\_theta(1) = V\_theta\_prime;

for i = 1:(length(theta\_s) - 1)

f1 = @(theta\_s, V\_r, V\_theta) V\_theta;

f2 = @(theta\_s, V\_r, V\_theta) (V\_theta^2 \* V\_r - (gamma - 1) / 2 \* (1 - V\_r^2 - V\_theta^2) \* (2 \* V\_r + V\_theta \* cot(theta\_s))) / ((gamma - 1) / 2 \* (1 - V\_r^2 - V\_theta^2) - V\_theta^2);

k11 = h \* f1(theta\_s(i), V\_r(i), V\_theta(i));

k12 = h \* f2(theta\_s(i), V\_r(i), V\_theta(i));

k21 = h \* f1(theta\_s(i) + 0.5 \* h, V\_r(i) + 0.5 \* k11, V\_theta(i) + 0.5 \* k12);

k22 = h \* f2(theta\_s(i) + 0.5 \* h, V\_r(i) + 0.5 \* k11, V\_theta(i) + 0.5 \* k12);

k31 = h \* f1(theta\_s(i) + 0.5 \* h, V\_r(i) + 0.5 \* k21, V\_theta(i) + 0.5 \* k22);

k32 = h \* f2(theta\_s(i) + 0.5 \* h, V\_r(i) + 0.5 \* k21, V\_theta(i) + 0.5 \* k22);

k41 = h \* f1(theta\_s(i) + h, V\_r(i) + k31, V\_theta(i) + k32);

k42 = h \* f2(theta\_s(i) + h, V\_r(i) + k31, V\_theta(i) + k32);

V\_r(i + 1) = V\_r(i) + (1 / 6) \* (k11 + 2 \* k21 + 2 \* k31 + k41);

V\_theta(i + 1) = V\_theta(i) + (1 / 6) \* (k12 + 2 \* k22 + 2 \* k32 + k42);

end

end