TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

An assignment report on determination of condition for formation of Mach stem using shock polar plot

Submitted by:

Sandip Adhikari

076bas035

Submitted to:

Asst. Professor Sudip Bhattarai

Department of Aerospace and Mechanical Engineering,

IOE, Pulchowk Campus

February 7, 2024

Introduction

Shock polar, also known as pressure-deflection shock polar, is the graphical representation of the pressure ratio (between flow downstream and upstream of an oblique shockwave) and the flow deflection angle, θ, for a given upstream Mach number. The plot represents the locus of all the possible flow states after an oblique shockwave formation. Shock polars are useful in understanding shock reflection phenomena as they enable the analysis of reflection patterns and prediction of transitions (from regular to irregular reflection).

Relations used to draw the plot are mentioned below:

a. Possible range of shockwave angle, $\boldsymbol{\beta}$

The minimum possible value of β corresponds to the value for Mach wave, which is equal to $\sin^{-1}\left(\frac{1}{M_1}\right)$. Here, M_1 is the upstream flow Mach number. And the maximum possible value corresponds to the value for normal shock wave; $\frac{\pi}{2}$ or 90°.

Hence, the range of β is given by

$$\sin^{-1}\left(\frac{1}{M_1}\right) < \beta < \frac{\pi}{2}$$

b. $\boldsymbol{\theta - \beta - M}$ relation for oblique shockwave

$$\tan\theta \;=\; 2\cot\beta\,\frac{M_1^2\, sin^2\beta - 1}{M_1^2(\gamma + \cos 2\beta) + 2}$$

Where,

θ = deflection angle; represents the deflection of the flow across the shockwave

$\gamma \;=\; \frac{c_p}{c_v}$ = ratio of specific heats

c. Pressure Ratio

$$\frac{P_2}{P_1} = 1 + \frac{2\gamma}{\gamma + 1}\,(M_1^2\, sin^2\beta \;-\; 1)$$

Here, P_1 and P_2 are the pressures upstream and downstream the oblique shockwave.

Upstream Flow and Wedge Conditions

Both the wedge angles θ_1 and θ_2 are taken to be 15° (as per sign convention, θ_1 is considered negative) whereas the upstream flow Mach number, M_1 is equal to 5.5. Figure 1 shows the representation of the conditions set.

The condition for the formation of Mach stem is determined from the shock polar plot, through trail and error, by keeping θ_1 constant and varying θ_2 .

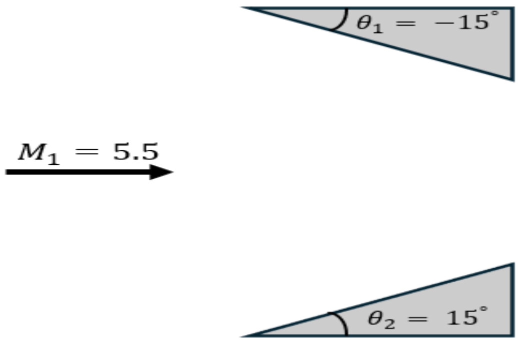

Figure 1: Representation of the upstream flow and initial wedge conditions

Plot

Figure 2 shows the baseline plot, which represents the locus of all the possible flow states after an oblique shockwave formation for $M_1 = 5.5$. Along with the baseline plot, it shows the possible states for starting wedge angles $\theta_1 = -15^{\circ}$ and $\theta_2 = 15^{\circ}$.

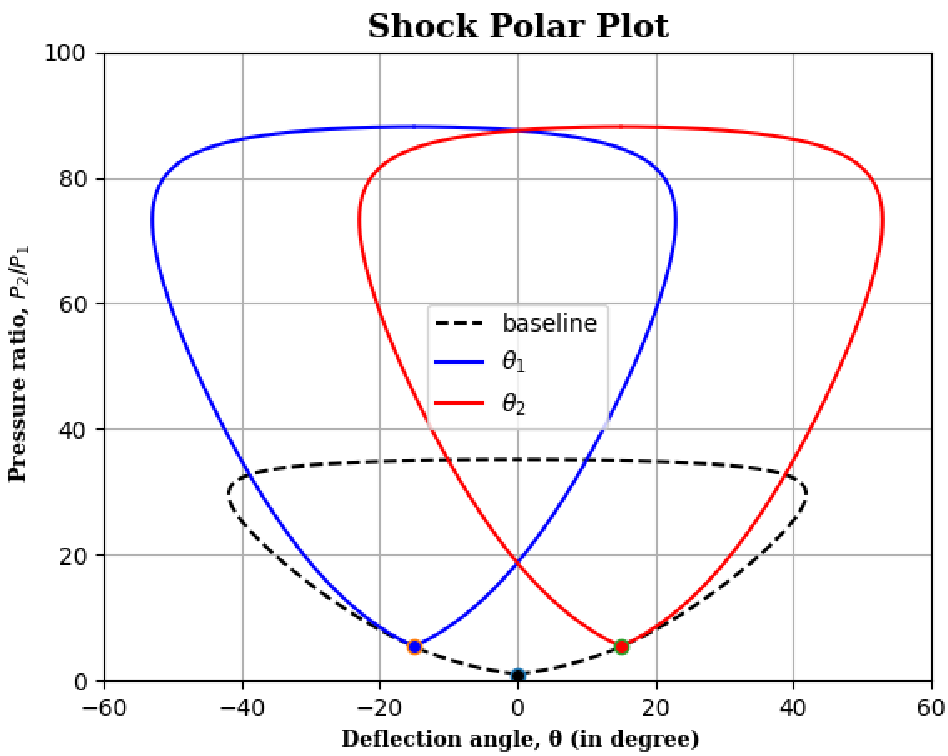

Figure 2: Shock Polar Plot for $\theta_1 = -15^{\circ}$ and $\theta_2 = 15^{\circ}$

On keeping the θ_1 constant and varying the θ_2, the two curves just separated (no intersection between curves) at $\theta_2 = 36.4^{\circ}$ (correct to one decimal place), which is shown in Figure 4, signifying the formation of Mach stem. Because the curves do not intersect, there is no common reflection point for the oblique shockwaves generated at the two wedges separately. So, in that case, Mach reflection occurs, resulting in Mach stem.

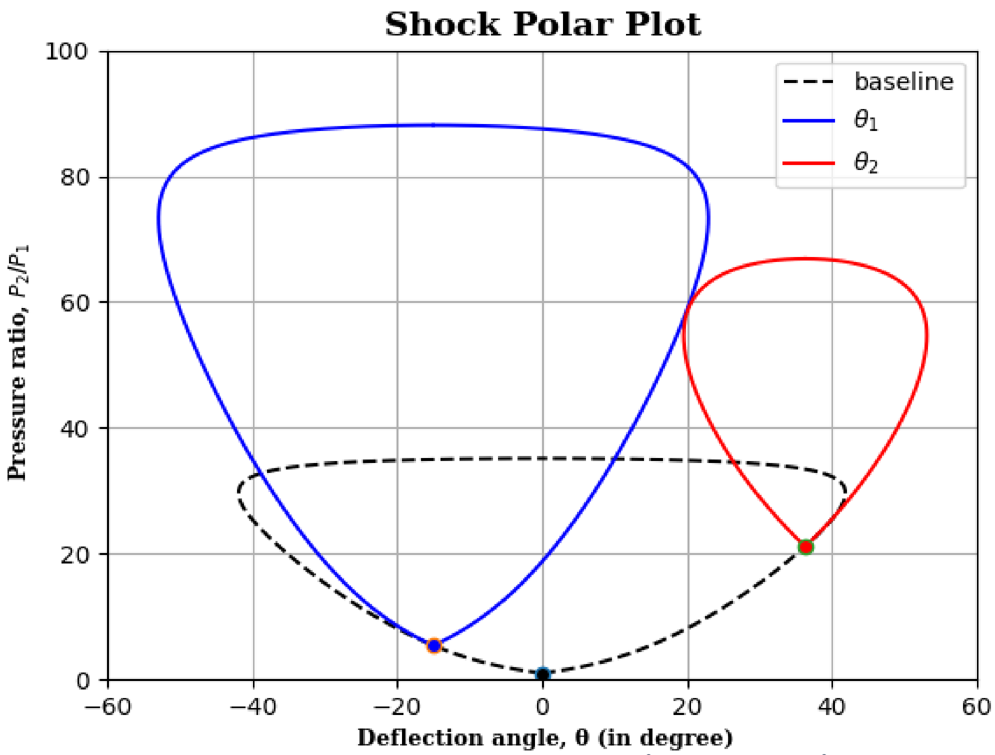

Figure 3: Shock Polar Plot for $\theta_1 = -15^{\circ}$ and $\theta_2 = 36.3^{\circ}$

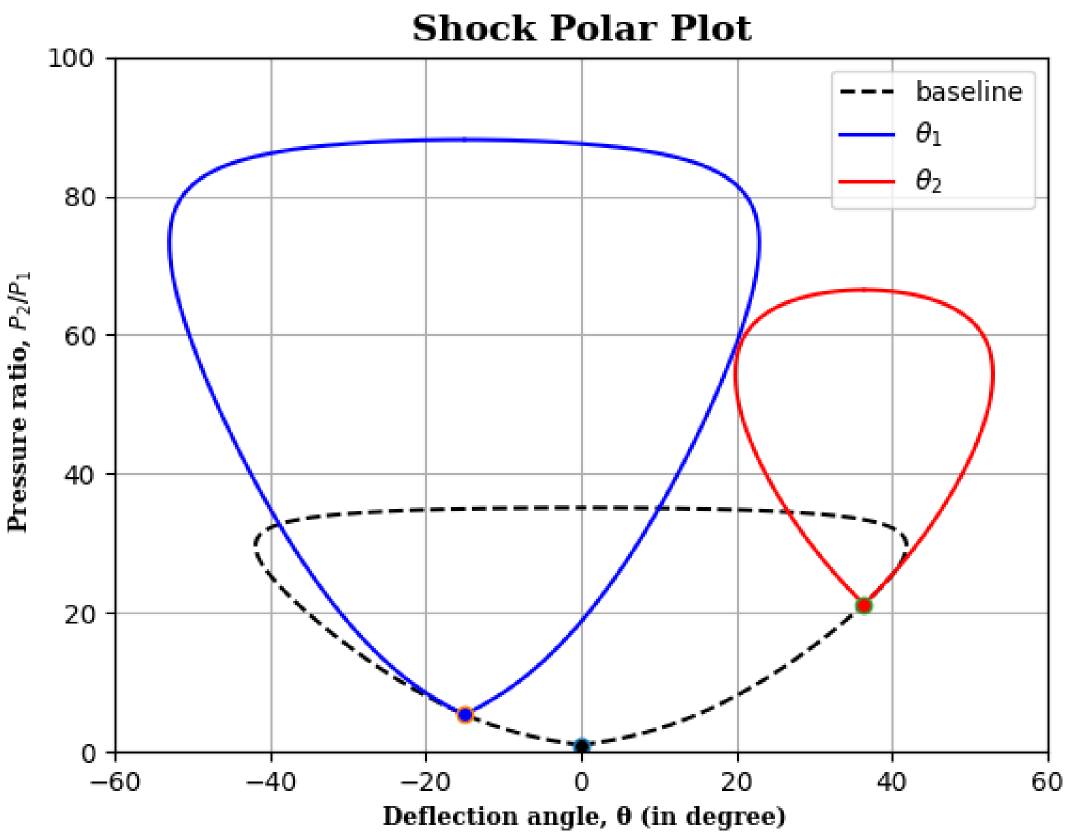

Figure 4: Shock Polar Plot for $\theta_1 = -15^{\circ}$ and $\theta_2 = 36.4^{\circ}$

Conclusion

A python code to plot the shock polar was written and compiled in VS Studio Code. And through the plot, the condition for the formation of Mach stem, in two wedge configurations, was determined.

APPENDIX I

Python Source Code:

```
# "All that had gone before was not a thousandth of what was yet to come; the
story of this star had barely begun."

import numpy as np
import scipy.optimize as optimize
import matplotlib.pyplot as plt

g = 1.4 # specific heat ratio (gamma)
M = 5.5 # upstream mach number
theta_1 = 15 # negative angle
theta_2 = 36.4

# Possible range of beta and theta for M | Baseline plot | locus of all the
possible shock wave formations for M
beta = np.arange(np.arcsin(1/M), np.pi/2, np.pi/10000)
theta = np.rad2deg(np.arctan(2*(M**2*np.sin(beta)**2-
1)/(np.tan(beta)*(M**2*(g+np.cos(2*beta))+2))))
PR = 1 + (2*g)/(g+1)*(M**2*np.sin(beta)**2-1)
plt.plot(theta, PR, 'k--', label='baseline' )
plt.plot(-theta, PR, 'k--', 0, 1, "o", markerfacecolor='k', linewidth=1.5)

#Function to find beta for a given theta
def calculate_beta(theta_val):
    theta_function = lambda beta_val,
theta_val:np.arctan(2*(M**2*np.sin(beta_val)**2-
1)/(np.tan(beta_val)*(M**2*(g+np.cos(2*beta_val))+2)))*180/np.pi - theta_val
    beta_initial_guess = 0
    beta_solution = optimize.fsolve(theta_function, beta_initial_guess,
args=(theta_val,))
    return beta_solution[0]

#Calculation of flow and shock wave properties for theta_1
beta_1 = calculate_beta(theta_1)
print(np.rad2deg(beta_1))

Mn1_1 = M * np.sin(beta_1)
Mn2_1 = np.sqrt((1+(g-1)/2*Mn1_1**2)/(g*Mn1_1**2-(g-1)/2))
M2_1 = Mn2_1/np.sin(beta_1-np.deg2rad(theta_1))
PR_1 = 1 + (2*g)/(g+1)*(M**2*np.sin(beta_1)**2-1)

beta1 = np.arange(np.arcsin(1/M2_1), np.pi/2, np.pi/10000)
theta1 = np.rad2deg(np.arctan(2*(M2_1**2*np.sin(beta1)**2-
1)/(np.tan(beta1)*(M2_1**2*(g+np.cos(2*beta1))+2))))
PR1 = (1 + (2*g)/(g+1)*(M2_1**2*np.sin(beta1)**2-1))*PR_1
plt.plot(-(theta_1+theta1), PR1, 'b', label = '$θ_{1}$')
```

```python
plt.plot(-(theta_1-theta1), PR1, 'b',-theta_1, PR_1, "o", markerfacecolor='b',
linewidth=1.5)

#Calculation of flow and shock wave properties for theta_2
beta_2 = calculate_beta(theta_2)
print(np.rad2deg(beta_2))

Mn1_2 = M * np.sin(beta_2)
Mn2_2 = np.sqrt((1+(g-1)/2*Mn1_2**2)/(g*Mn1_2**2-(g-1)/2))
M2_2 = Mn2_2/np.sin(beta_2-np.deg2rad(theta_2))
PR_2 = 1 + (2*g)/(g+1)*(M**2*np.sin(beta_2)**2-1)

beta2 = np.arange(np.arcsin(1/M2_2), np.pi/2, np.pi/10000)
theta2 = np.rad2deg(np.arctan(2*(M2_2**2*np.sin(beta2)**2-
1)/(np.tan(beta2)*(M2_2**2*(g+np.cos(2*beta2))+2))))
PR2 = (1 + (2*g)/(g+1)*(M2_2**2*np.sin(beta2)**2-1))*PR_2
plt.plot(theta_2+theta2, PR2, 'r',label = '$θ_{2}$')
plt.plot(theta_2-theta2, PR2, 'r', theta_2, PR_2, "o", markerfacecolor='r',
linewidth=1.5)

# Set font properties for labels
font_propl = {
    'family': 'serif',
    'size': 9,
    'weight': 'bold'
}
# Set font properties for title
font_propt = {
    'family': 'serif',
    'size': 14,
    'weight': 'bold'
}

# Set plot title and labels
plt.title('Shock Polar Plot', fontdict = font_propt)
plt.xlabel(' Deflection angle, θ (in degree)', fontdict = font_propl)
plt.ylabel(' Pressure ratio, $P_{2}$/$P_{1}}$', fontdict = font_propl)
plt.xlim([-60, 60])
plt.ylim([0, 100])
plt.grid(True)
plt.legend()
plt.show()
```