AE339
Flow Parameter Calculator Report

Anuttar Jain
22B0003

October 15^{th} 2024

Aim

To calculate various flow parameters, with any one input parameter, under 3 scenarios-

- Isentropic Flow
- Normal Shock
- Oblique Shock

Isentropic Flow Parameters:

- Mach Number, M
- Mach Angle, μ
- Prandtl-Mayer Angle, ν
- P/P0
- T/T0
- $\rho/\rho 0$
- A/A^*
- P/P^*
- T/T^*
- ρ/ρ^*

Normal Shock Parameters:

- M1
- M2
- P2/P1
- T2/T1
- $\rho 2/\rho 1$
- P02/P01
- P1/P02

Oblique Shock Parameters:

- M1
- M1n
- M2
- M2n
- Wave Angle, β
- Turn Angle, δ
- P2/P1
- T2/T1
- $\rho 2/\rho 1$
- P02/P01

Theory

Isentropic Flow

Isentropic 1D (and Quasi 1D) Flow Relations in terms of Mach Number:

$$\frac{P}{P_0} = (1 + \frac{\gamma - 1}{2} M^2)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{T}{T_0} = (1 + \frac{\gamma - 1}{2} M^2)$$

$$\frac{\rho}{\rho_0} = (1 + \frac{\gamma - 1}{2} M^2)^{\frac{1}{\gamma-1}}$$

$$\mu = sin^{-1}(\frac{1}{M})$$

$$\nu = \sqrt{\frac{\gamma + 1}{\gamma - 1}} tan^{-1} \sqrt{\frac{\gamma - 1}{\gamma + 1}(M^2 - 1)} - tan^{-1} \sqrt{M^2 - 1}$$

$$\frac{A}{A^*} = \frac{1}{M} \left(\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

$$\frac{P}{P^*} = \left(\frac{(\gamma + 1)/2}{1 + (\gamma + 1) M^2/2} \right)^{\gamma/(\gamma-1)}$$

$$\frac{T}{T^*} = \frac{(\gamma + 1)/2}{1 + (\gamma + 1) M^2/2}$$

$$\frac{\rho}{\rho^*} = \left(\frac{(\gamma + 1)/2}{1 + (\gamma + 1) M^2/2} \right)^{1/(\gamma-1)}$$

Expression for Mach Number using other Isentropic relations:

If P/P0 is given:

$$M = \sqrt{\left(\left(\frac{P}{P_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) \left(\frac{2}{\gamma - 1} \right)}$$

If T/T0 is given:

$$M = \sqrt{\left(\frac{T}{T_0} - 1 \right) \left(\frac{2}{\gamma - 1} \right)}$$

If $\rho/\rho0$ is given:

$$M = \sqrt{\left(\left(\frac{\rho}{\rho_0} \right)^{\gamma-1} - 1 \right) \left(\frac{2}{\gamma - 1} \right)}$$

If μ is given:

$$M = \frac{1}{sin(\mu)}$$

Normal Shock Relations

Normal Shock Relations in terms of incoming Mach Number, M1:

$$M_2 = \sqrt{\frac{(\gamma-1)M_1^2+2}{2\gamma M_1^2-(\gamma-1)}}$$

$$\frac{T_2}{T_1} = \frac{(2\gamma M_1^2-(\gamma-1))(2+(\gamma-1)M_1^2)}{(\gamma+1)^2 M_1^2}$$

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

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)M_1^2}{2+(\gamma-1)M_1^2}$$

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

$$\frac{P_1}{P_{02}} = \frac{((\gamma+1)M_1^2/2)^{\gamma/(\gamma-1)}}{\left(\frac{2\gamma M_1^2}{\gamma+1}-\frac{\gamma-1}{\gamma+1}\right)^{1/(\gamma-1))}}$$

Oblique Shock Relations

Oblique Shock Relations in terms of Mach Number, M1 and Wave Angle, β:

$$M_{1n} = M_1 sin\beta$$

$$tan\delta = \frac{2cot\beta(M_1^2 sin^2\beta-1)}{2+M_1^2(\gamma+cos2\beta)}$$

$$M_2 = \frac{M_{2n}}{sin(\beta-\delta)}$$

Code

A python class: calculator, is created to evaluate the flow parameters, given an input, and present them in a tabular format.

Isentropic Flow Functions

- **isentropic_Mach (self, M):** all the parameters are evaluated using the equations given above and are presented in a tabular form.
- **isentropic_pp0 (self, pp0):** pp0 is P/P0, input parameter. Mach number, M is evaluated using P/P0, and all other parameters are calculated using isentropic_Mach() function.
- **isentropic_TT0 (self, tt0):** tt0 is T/T0, input parameter. Mach number, M is evaluated using T/T0, and all other parameters are calculated using isentropic_Mach() function.
- **isentropic_RhoRho0 (self, rr0):** rr0 is $\rho/\rho0$, input parameter. Mach number, M is evaluated using $\rho/\rho0$, and all other parameters are calculated using isentropic_Mach() function.

- **isentropic_MachAngle (self, mu):** mu is Mach Angle μ, input parameter. Mach number, M is evaluated using μ, and all other parameters are calculated using isentropic_Mach() function.
- **isentropic_PMAngle (self, nu):** nu is Prandtl-Mayer Angle ν, input parameter. The function f(self, M) is the function that evaluates ν - ν_0 taking M as input. Using this function and a given ν_0 angle, M is evaluated using fsolve() function from scipy.optimize library by finding the root of f() function. All other parameters are calculated using isentropic_Mach() function.
- **isentropic_AAstar_subsonic (self, aastar):** f2(self, M) is a function that evaluates A/Astar - $A/Astar_0$ from input M, where $A/Astar_0$ is a certain input A/Astar value. Now, required M is calculated fsolve() over f2() in the subsonic regime. Remaining parameters are calculated using isentropic_Mach() function.
- **isentropic_AAstar_supersonic (self, aastar):** f2(self, M) is a function that evaluates A/Astar - $A/Astar_0$ from input M, where $A/Astar_0$ is a certain input A/Astar value. Now, required M is calculated fsolve() over f2() in the supersonic regime. Remaining parameters are calculated using isentropic_Mach() function.

Normal Shock Functions

- **normalShock_M1(self, M1):** all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_M2(self, M2):** M1 is calculated from M2 first. Then, all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_P2P1(self, p2p1):** P2P1 is P2/P1. M1 is calculated from P2/P1 first. Then, all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_Rho2Rho1(self, r2r1):** r2r1 is $\rho 2/\rho 1$. M1 is calculated from r2r1 first. Then, all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_T2T1(self, t2t1):** t2t1 is T2/T1. g1(self, M1) is a function that finds T2/T1 - T2/T1 _ 0, taking M1 as input. So, for a given T2/T1 _ 0, we can find corresponding M1 using fsolve() function. Then, all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_P02P01(self, p02p01):** p02p01 is P02/P01. g2(self, M1) is a function that finds P02/P01 - P02/P01 _ 0, taking M1 as input. So, for a given P02/P01 _ 0, we can find corresponding M1 using fsolve() function. Then, all the other parameters are evaluated using their equations and presented in a tabular form.
- **normalShock_P1P02(self, p1p02):** p1p02 is P1/P02. g3(self, M1) is a function that finds P1/P02 - P1/P02 _ 0, taking M1 as input. So, for a given P1/P02 _ 0, we can find corresponding M1 using fsolve() function. Then, all the other parameters are evaluated using their equations and presented in a tabular form.

Oblique Shock Functions

For oblique shocks, Normal Mach component is evaluated as M1n first using a given Wave angle. Then using normal shock relations, the rest components are evaluated. M2 is M2n here. The actual M2 is calculated from M2n from equation given above.

- **obliqueShock_WaveAngle (self, M1, B):** B is Wave angle, β. The function just evaluates parameters as described above, and returns them in a tabular form.
- **obliqueShock_M1n (self, M1, M1n):** Wave angle is calculated from M1n and M1. Then all the remaining parameters are calculated using the obliqueShock_WaveAngle() function.
- **obliqueShock_TurnAngle_weak (self, M1, delta):** delta here is the turn angle δ. h1(self, B) is a function that evaluates δ - $\delta 0$ using β. Now given a delta, we can reverse calculate smaller value of beta (for weak shocks) using fsolve() function. Known beta and M1, all other parameters are calculated.
- **obliqueShock_TurnAngle_strong (self, M1, delta):** delta here is the turn angle δ. h1(self, B) is a function that evaluates δ - $\delta 0$ using β. Now given a delta, we can reverse calculate larger value of beta (for strong shocks) using fsolve() function. Known beta and M1, all other parameters are calculated.