```python
import numpy as np
from scipy.optimize import fsolve
from scipy.integrate import odeint

def theta_beta_mach(beta, M1, gamma, theta):
    left = 2 * np.tan(beta) * (M1**2 * np.sin(beta)**2 - 1)
    right = M1**2 * (gamma + np.cos(2*beta)) + 2
    return np.arctan(left / right) - theta

def normal_shock_M2(M1n, gamma):
    numerator = 1 + ((gamma-1)/2)*M1n**2
    denominator = gamma*M1n**2 - (gamma-1)/2
    return np.sqrt(numerator / denominator)

def taylor_maccoll(y, theta, gamma=1.4):
    v_r, v_theta = y
    denom = (gamma - 1) / 2 * (1 - v_r ** 2 - v_theta ** 2) - v_theta ** 2
    if np.abs(denom) < 1e-8: denom = 1e-8
    deriv_vr = v_theta
    deriv_vtheta = (
        v_theta ** 2 * v_r
        - (gamma - 1) / 2 * (1 - v_r ** 2 - v_theta ** 2)
        * (2 * v_r + v_theta / np.tan(theta))
    ) / denom
    return [deriv_vr, deriv_vtheta]

def cone_shock(cone_angle_deg, M1, gamma=1.4, N=1000):

    theta = np.deg2rad(cone_angle_deg)

    beta_guess = [theta + np.deg2rad(1), np.deg2rad(80)]

    beta_sol = fsolve(theta_beta_mach, np.mean(beta_guess), args=(M1, gamma, theta))[0]

    M1n = M1 * np.sin(beta_sol)
```

```python
    M2n = normal_shock_M2(M1n, gamma)
    # Flow behind the shock
    VR0 = M2n / M1    # Nondimensional radial velocity
    Vtheta0 = 0       # At the shock, all velocity change is normal
    y0 = [VR0, Vtheta0]
    thetas = np.linspace(beta_sol, theta, N)
    sol = odeint(taylor_maccoll, y0, thetas, args=(gamma,))

    vtheta_surface = sol[-1, 1]
    return {
        "shock_angle_deg": np.rad2deg(beta_sol),
        "beta_rad": beta_sol,
        "M1n": M1n,
        "M2n": M2n,
        "cone_surface_vtheta": vtheta_surface,
        "thetas": thetas,
        "Vr": sol[:,0],
        "Vtheta": sol[:,1],
    }

# inputs here
############################################
cone_angle_deg = 35
M1 = 2
results = cone_shock(cone_angle_deg, M1)
############################################


print(f"Shock angle: {results['shock_angle_deg']:.2f} deg")
print(f"Normal Mach behind shock: {results['M2n']:.4f}")
print(f"Vtheta at cone surface (should be 0): {results['cone_surface_vtheta']:.4e}")

import matplotlib.pyplot as plt
plt.plot(np.rad2deg(results['thetas']), results['Vr'], label="Vr")
```

```python
plt.plot(np.rad2deg(results['thetas']), results['Vtheta'], label="Vtheta")
plt.xlabel("theta (deg)")
plt.ylabel("Velocity (nondimensional)")
plt.legend()
plt.title("Taylor-Maccoll velocities from shock to cone surface")
plt.show()


```