Table of Contents

HYPERSONIC FLOWS AE 625

```
%VITTORIO BARALDI - HOMEWORK 5
clear all
close all
```

Integrate the Taylor-Maccoll Equation

```
waveangle1=[13:1:78];
n=length(waveangle1);%wave angle
thetashock=waveangle1*pi/180; %wave angle in radians
g=1.4;%heat air coefficient
f=@(x,y,z) (z^2*y-((g-1)/2)*(1-y^2-z^2)*(2*y+z*cot(x)))/...
    (((g-1)/2)*(1-y^2-z^2)-z^2);
thetashock2=zeros(n,n);


%small increment
dt=-0.001;
%tolerance
tol=0.003;
```

M = 5

```
M1=5; %mach numbers

 for i=1:n

    for j=1:500
        if j==1
        %calculating deflection angle and mach no. after shock
        delta1(i,j)= atan(2*cot(thetashock(1,i))*...
            (((M1^2*sin(thetashock(1,i))^2)-1)/(M1^2*...
            (g+cos(2*thetashock(1,i)))+2)));
        M2_1(i,j)=(1/(sin(thetashock(1,i)-delta1(i,j))))*...
            sqrt((1+((g-1)/2)*M1^2*sin(thetashock(1,i))^2)/...
                        ((g*M1^2*sin(thetashock(1,i))^2)-((g-1)/2)));
        %non-dimensional velocity
        v(i,j)=((2/((g-1)*M2_1(i,j)^2))+1)^(-1/2);
        %radial and tangential non-dimensional velocity components
```

```
        vr(i,j)=v(i,j)*cos(thetashock(1,i)-delta1(i,j));
        vtheta(i,j)=v(i,j)*sin(thetashock(1,i)-delta1(i,j));
        vr0(i,j)=-vtheta(i,j);

        thetashock2(i,j)=thetashock(1,i);

        %Runge-Kutta integration (1st step)
        m1(i,j)=dt*f(thetashock2(i,j),vr(i,j),vr0(i,j));
        k1(i,j)=dt*vr0(i,j);
        m2(i,j)=dt*f(thetashock2(i,j)+dt/2,vr(i,j)+...
            k1(i,j)/2,vr0(i,j)+m1(i,j)/2);
        k2(i,j)=dt*(vr0(i,j)+m2(i,j)/2);
        m3(i,j)=dt*f(thetashock2(i,j)+dt/2,vr(i,j)+...
            k2(i,j)/2,vr0(i,j)+m2(i,j)/2);
        k3(i,j)=dt*(vr0(i,j)+m3(i,j)/2);
        m4(i,j)=dt*f(thetashock2(i,j)+dt,vr(i,j)+...
            k3(i,j),vr0(i,j)+m3(i,j));
        k4(i,j)=dt*(vr0(i,j)+m4(i,j));

        %new values for Vr and Vtheta
        vr0(i,j
+1)=vr0(i,j)+((m1(i,j)+2*m2(i,j)+2*m3(i,j)+2*m4(i,j))/6);
        vr(i,j+1)=vr(i,j)+((k1(i,j)+2*k2(i,j)+2*k3(i,j)+2*k4(i,j))/6);
        thetashock2(i,j+1)=thetashock2(i,j)+dt;
        else

        %runge-kutta integration (next steps)
        m1(i,j)=dt*f(thetashock2(i,j),vr(i,j),vr0(i,j));
        k1(i,j)=dt*vr0(i,j);
        m2(i,j)=dt*f(thetashock2(i,j)+dt/2,vr(i,j)...
            +k1(i,j)/2,vr0(i,j)+m1(i,j)/2);
        k2(i,j)=dt*(vr0(i,j)+m2(i,j)/2);
        m3(i,j)=dt*f(thetashock2(i,j)+dt/2,vr(i,j)...
            +k2(i,j)/2,vr0(i,j)+m2(i,j)/2);
        k3(i,j)=dt*(vr0(i,j)+m3(i,j)/2);
        m4(i,j)=dt*f(thetashock2(i,j)+dt,vr(i,j)...
            +k3(i,j),vr0(i,j)+m3(i,j));
        k4(i,j)=dt*(vr0(i,j)+m4(i,j));

        vr0(i,j
+1)=vr0(i,j)+((m1(i,j)+2*m2(i,j)+2*m3(i,j)+2*m4(i,j))/6);
        vr(i,j+1)=vr(i,j)+((k1(i,j)+2*k2(i,j)+2*k3(i,j)+2*k4(i,j))/6);

        %check on Vtheta to be equal to zero (or close)
        if abs(vr0(i,j+1))<=tol
            if (thetashock2(i,j))>0
            %cone angle for each iteration
            thetacone1(i)=thetashock2(i,j)+dt;
            break
            end
        else
            thetashock2(i,j+1)=thetashock2(i,j)+dt;
        end
      end
```

```
        end
    end
```

M = 10

```
    %repeating above procedure for mach 10 and mach 15
   M2=10;
   waveangle2=[5:1:78];
   n=length(waveangle2);
   thetashock=waveangle2*pi/180;
   thetashock3=zeros(n,n);
   thetashock4=zeros(n,n);
    for i=1:n

        for j=1:500
            if j==1

            delta2(i,j)= atan(2*cot(thetashock(1,i))*(((M2^2*...
                sin(thetashock(1,i))^2)-1)/(M2^2*...
                         (g+cos(2*thetashock(1,i)))+2)));
            M2_2(i,j)=(1/(sin(thetashock(1,i)-delta2(i,j))))*...
                sqrt((1+((g-1)/2)*M2^2*sin(thetashock(1,i))^2)/...
                            ((g*M2^2*sin(thetashock(1,i))^2)-((g-1)/2)));
            v2(i,j)=((2/((g-1)*M2_2(i,j)^2))+1)^(-1/2);
            vr_2(i,j)=v2(i,j)*cos(thetashock(1,i)-delta2(i,j));
            vtheta2(i,j)=v2(i,j)*sin(thetashock(1,i)-delta2(i,j));
            vr0_2(i,j)=-vtheta2(i,j);
            thetashock3(i,j)=thetashock(1,i);
            m1_2(i,j)=dt*f(thetashock3(i,j),vr_2(i,j),vr0_2(i,j));
            k1_2(i,j)=dt*vr0_2(i,j);
            m2_2(i,j)=dt*f(thetashock3(i,j)+dt/2,vr_2(i,j)+...
                k1_2(i,j)/2,vr0_2(i,j)+m1_2(i,j)/2);
            k2_2(i,j)=dt*(vr0_2(i,j)+m2_2(i,j)/2);
            m3_2(i,j)=dt*f(thetashock3(i,j)+dt/2,vr_2(i,j)+...
                k2_2(i,j)/2,vr0_2(i,j)+m2_2(i,j)/2);
            k3_2(i,j)=dt*(vr0_2(i,j)+m3_2(i,j)/2);
            m4_2(i,j)=dt*f(thetashock3(i,j)+dt,vr_2(i,j)+...
                k3_2(i,j),vr0_2(i,j)+m3_2(i,j));
            k4_2(i,j)=dt*(vr0_2(i,j)+m4_2(i,j));
            vr0_2(i,j+1)=vr0_2(i,j)+((m1_2(i,j)+2*...
                m2_2(i,j)+2*m3_2(i,j)+2*m4_2(i,j))/6);
            vr_2(i,j+1)=vr_2(i,j)+((k1_2(i,j)+2*...
                k2_2(i,j)+2*k3_2(i,j)+2*k4_2(i,j))/6);
            thetashock3(i,j+1)=thetashock3(i,j)+dt;

            else

            m1_2(i,j)=dt*f(thetashock3(i,j),vr_2(i,j),vr0_2(i,j));
            k1_2(i,j)=dt*vr0_2(i,j);
            m2_2(i,j)=dt*f(thetashock3(i,j)+dt/2,vr_2(i,j)+...
                k1_2(i,j)/2,vr0_2(i,j)+m1_2(i,j)/2);
            k2_2(i,j)=dt*(vr0_2(i,j)+m2_2(i,j)/2);
            m3_2(i,j)=dt*f(thetashock3(i,j)+dt/2,vr_2(i,j)+...
```

```
            k2_2(i,j)/2,vr0_2(i,j)+m2_2(i,j)/2);
        k3_2(i,j)=dt*(vr0_2(i,j)+m3_2(i,j)/2);
        m4_2(i,j)=dt*f(thetashock3(i,j)+dt,vr_2(i,j)+...
            k3_2(i,j),vr0_2(i,j)+m3_2(i,j));
        k4_2(i,j)=dt*(vr0_2(i,j)+m4_2(i,j));
        vr0_2(i,j
+1)=vr0_2(i,j)+((m1_2(i,j)+2*m2_2(i,j)+2*m3_2(i,j)+2*...
            m4_2(i,j))/6);
        vr_2(i,j
+1)=vr_2(i,j)+((k1_2(i,j)+2*k2_2(i,j)+2*k3_2(i,j)+2*...
            k4_2(i,j))/6);

        if abs(vr0_2(i,j+1))<=tol
            thetacone2(i)=thetashock3(i,j)+dt;
            break
        else
            thetashock3(i,j+1)=thetashock3(i,j)+dt;
        end
        end
    end
end
```

M = 15

```
M3=15;

for i=1:n
    for j=1:500
        if j==1
        delta3(i,j)= atan(2*cot(thetashock(1,i))*...
            (((M3^2*sin(thetashock(1,i))^2)-1)/(M3^2*...
                     (g+cos(2*thetashock(1,i)))+2)));
        M2_3(i,j)=(1/(sin(thetashock(1,i)-delta3(i,j))))*...
            sqrt((1+((g-1)/2)*M3^2*sin(thetashock(1,i))^2)/...
                        ((g*M3^2*sin(thetashock(1,i))^2)-((g-1)/2)));
        v3(i,j)=((2/((g-1)*M2_3(i,j)^2))+1)^(-1/2);
        vr_3(i,j)=v3(i,j)*cos(thetashock(1,i)-delta3(i,j));
        vtheta3(i,j)=v3(i,j)*sin(thetashock(1,i)-delta3(i,j));
        vr0_3(i,j)=-vtheta3(i,j);
        thetashock4(i,j)=thetashock(1,i);
        m1_3(i,j)=dt*f(thetashock4(i,j),vr_3(i,j),vr0_3(i,j));
        k1_3(i,j)=dt*vr0_3(i,j);
        m2_3(i,j)=dt*f(thetashock4(i,j)+dt/2,vr_3(i,j)+...
            k1_3(i,j)/2,vr0_3(i,j)+m1_3(i,j)/2);
        k2_3(i,j)=dt*(vr0_3(i,j)+m2_3(i,j)/2);
        m3_3(i,j)=dt*f(thetashock4(i,j)+dt/2,vr_3(i,j)+...
            k2_3(i,j)/2,vr0_3(i,j)+m2_3(i,j)/2);
        k3_3(i,j)=dt*(vr0_3(i,j)+m3_3(i,j)/2);
        m4_3(i,j)=dt*f(thetashock4(i,j)+dt,vr_3(i,j)+...
            k3_3(i,j),vr0_3(i,j)+m3_3(i,j));
        k4_3(i,j)=dt*(vr0_3(i,j)+m4_3(i,j));

        vr0_3(i,j+1)=vr0_3(i,j)+((m1_3(i,j)+2*m2_3(i,j)+...
```

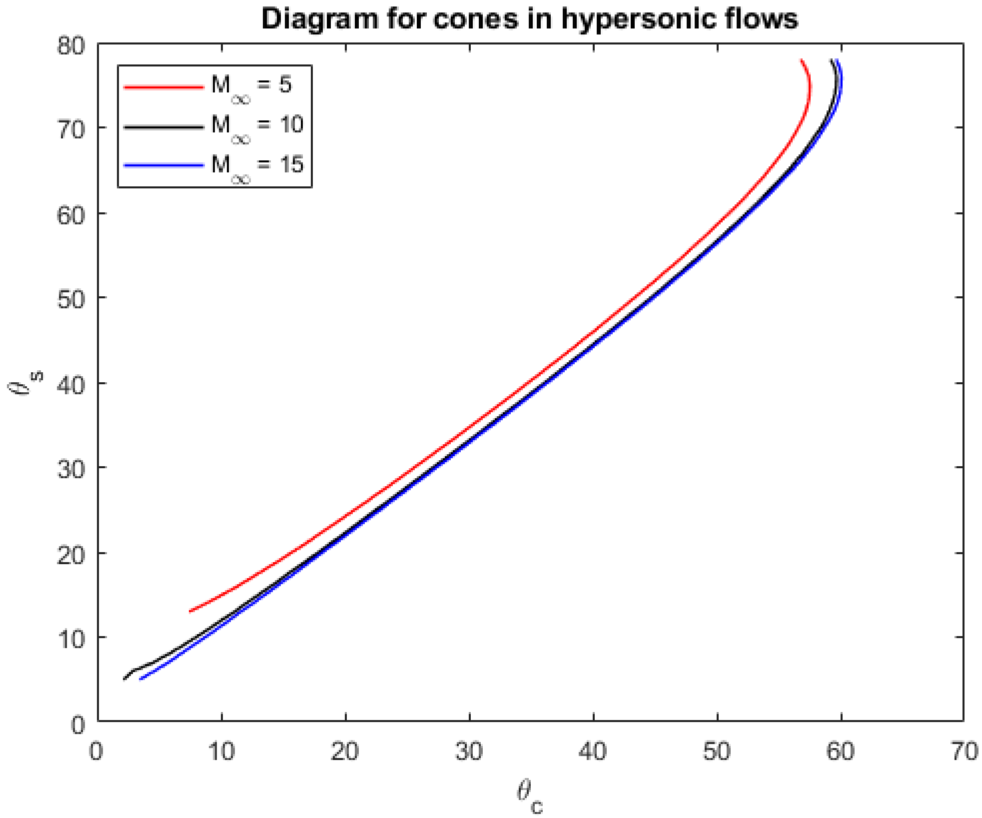
```
            2*m3_3(i,j)+2*m4_3(i,j))/6);
        vr_3(i,j+1)=vr_3(i,j)+((k1_3(i,j)+2*k2_3(i,j)+...
            2*k3_3(i,j)+2*k4_3(i,j))/6);
        thetashock4(i,j+1)=thetashock4(i,j)+dt;
        else
        m1_3(i,j)=dt*f(thetashock4(i,j),vr_3(i,j),vr0_3(i,j));
        k1_3(i,j)=dt*vr0_3(i,j);
        m2_3(i,j)=dt*f(thetashock4(i,j)+dt/2,vr_3(i,j)+...
            k1_3(i,j)/2,vr0_3(i,j)+m1_3(i,j)/2);
        k2_3(i,j)=dt*(vr0_3(i,j)+m2_3(i,j)/2);
        m3_3(i,j)=dt*f(thetashock4(i,j)+dt/2,vr_3(i,j)+...
            k2_3(i,j)/2,vr0_3(i,j)+m2_3(i,j)/2);
        k3_3(i,j)=dt*(vr0_3(i,j)+m3_3(i,j)/2);
        m4_3(i,j)=dt*f(thetashock4(i,j)+dt,vr_3(i,j)+...
            k3_3(i,j),vr0_3(i,j)+m3_3(i,j));
        k4_3(i,j)=dt*(vr0_3(i,j)+m4_3(i,j));
        vr0_3(i,j+1)=vr0_3(i,j)+((m1_3(i,j)+2*m2_3(i,j)+...
            2*m3_3(i,j)+2*m4_3(i,j))/6);
        vr_3(i,j+1)=vr_3(i,j)+((k1_3(i,j)+2*k2_3(i,j)+...
            2*k3_3(i,j)+2*k4_3(i,j))/6);

        if abs(vr0_3(i,j+1))<=tol
            thetacone3(i)=thetashock4(i,j)+dt;
        else
            thetashock4(i,j+1)=thetashock4(i,j)+dt;
        end
       end
    end
end
```

Plots

```
plot (thetacone1*180/pi,waveangle1,'-r','LineWidth',1.05)
hold on
plot(thetacone2*180/pi,waveangle2,'-k','LineWidth',1.05)
hold on
plot(thetacone3*180/pi,waveangle2,'-b','LineWidth',1.05)

title('Diagram for cones in hypersonic flows')
xlabel('\theta_c')
ylabel('\theta_s')
legend('M_\infty = 5','M_\infty = 10','M_\infty = 15', 'Location',...
    'northwest')
```



REPORT

```
%{
Objective: integrate Taylor-Maccoll equation using runge-kutta method.
Given a shock range of angles, find the corresponding cone angle for
 Mach
5, 10 and 15.

Procedure:
- initialize a vector for different shock wave angles (from 5° to
 70-80°,
conformed with literature tables)
- express our eqn 10.13 (from the handed out notes) in terms of
non-dimensional velocities (f(x,y,z) in this code)
- pick a tolerance and a small increment to apply to the integration
 (tol
and dt)
- using the oblique shock relations, find the mach number after shock
 and
determine the wedge angle for that shock wave (here called 'delta')
- find the normal and the tangential component of the velocity after
 the
shock (non-dimensionalized)
- integrate using runge-kutta procedure, calculating the K's and M's
coefficients. The wave angle, the normal velocity and the tangential
```

```
velocity are the initial conditions for the integration.
- using the apposite expressions, update the new values for Vtheta and
 Vr.
Check for Vtheta to be less than the tolerance (or zero)
- If Vtheta is zero, then the angle used for calculations on that
iteration corresponds to our cone angle, otherwise we can increment
 our
angle by dt again, and re-iterate
- once determined the cone angle for a specific initial wave angle,
 the
next wave angle is analyzed and its corresponding cone angle found
 with the
same procedure
- once every cone angle is determined for every wave angle of the
 array,
then the same piece of code is run for each Mach number considered
(5-10-15)
- Results are plotted with the shock angle on the y-axis and the cone
 angle
on the x-axis for each Mach no.


VITTORIO BARALDI
%}
```

Published with MATLAB® R2019b