ATTENTION!

This Page Is the Cover Page of Your Project, Do Not Put Anything Before this Page

MEEN-414-646 Elements of Turbomachinery , Module M 2

Design of Subsonic Turbine M2-T and Compressor Blades M2-C

No. of students involved: 2 for 1 for M-2T and 1 for M-2C
Student 1: Last Name, Initial
Student 2: Last Name, Initial

Distributed: 01/27/15
Final Date: 02/03/15

M-2T Subsonic Turbine Blade Design Instruction

For the turbine design project, develop a design software that enables you to generate subsonic turbine blades.

1) Read Chapter 10, turbine and compressor part, **you may omit the conformal transformation part**.
2) Generate a family of profiles that have $\alpha_1 = 90^o$, $\alpha_2 = 160^o$, $\alpha_1 = 45^o$, $\alpha_2 = 160^o$,
3) Superpose the turbine base profile on the camber using the code you already developed

To design subsonic turbine blades, a step-by-step design procedure is given, see also Fig. 1
Step1

1) Draw two parallel lines that are apart by an axial chord.
2) Choose an axial chord C_{ax} of any length you wish.
3) Draw at 1/3 of the C_{ax} a parallel line to the cascade front.
4) Draw the inlet flow velocity vector with the given angle α_1 and intersect it with the 1/3 line.
5) At the point of intersection draw exit velocity vector line at an angle α_2.
6) Intersect the above line with the cascade rear front line.
7) Connect the front intersect with the rear, you find the stagger angle γ
8) With the tangent at inlet and exit as Fig 1. Construct the camberline using **Bezier function** as detailed in Chapter 10. You may use instead of 1/3 a one half distance (item 3)
9) Once you have the camberline, superimpose the base profile as detailed in Chapter 10. The base profile is in ***ecampus*** as a data file.

Step 2: Manual design of camber line

1) Divide segment A and B into equal segments as shown in Fig.1.
2) Connect the point 2 of A with point n-1 of B, 3 with n-2, etc to design a camber envelope.
3) Tangent the camber to the envelope.
4) You may curve fit the camber or extract discrete points.
5) Use the camber in conjunction with a given base line. Use the superposition principle, Chap 10 .

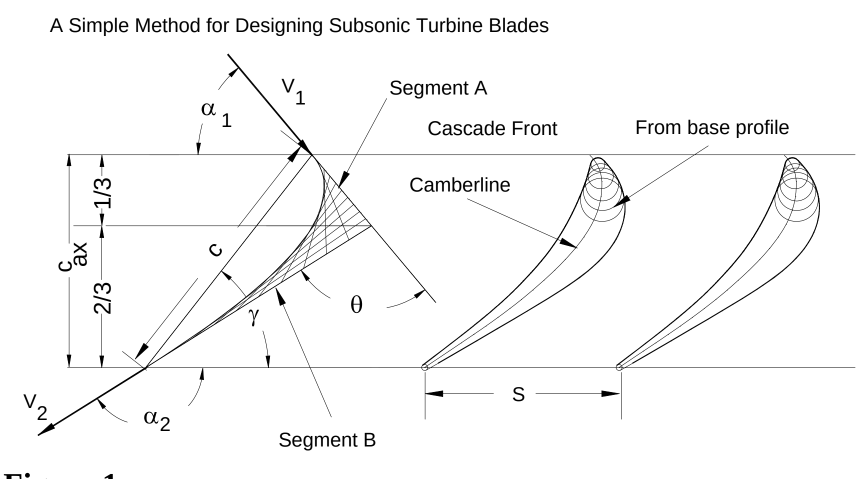


Figure 1

M2C Module: Subsonic Compressor Design Using NACA Camberline Equation

For the compressor design project, develop a design software that enables you to generate NACA compressor blade profiles

1) Read Chapter 10
2) Follow instructions and arrive at the equation for blade camber line for compressor
3) For compressor blades, the higher the blade camber height, the higher is the lift coefficient Cl
   Vary Cl from 0 to 1.0 to get a feeling about the camber.
4) Superimpose the following base profile and get the compressor blade
5) Further instruction will be given to you in the class.

For generation of suction and pressure side you may spline function or any other interpolation function.

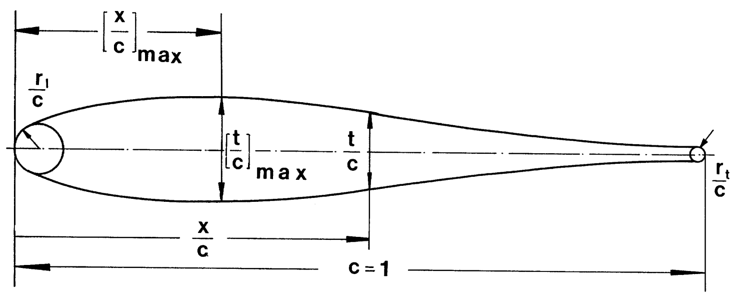


Figure 2: Base profile

$[x/c]_{max\%}$	5	10	15	20	25	30	35	40	45	50
$[t/c]_{max\%}$	8.4	11.1	13.0	14.4	14.9	15.0	14.4	13.5	12.4	11.2
$[x/c]_{max\%}$	55	60	65	70	75	80	85	90	95	100
$[t/c]_{max\%}$	10.1	8.9	7.7	6.7	5.7	4.7	3.8	3.0	2.1	0.0

Also: The following base profile generated by Joukosky transformation will be inserted into a special folder in ***ecampus*** that you have access to. Once you have the file take Column 1 and 2 only.

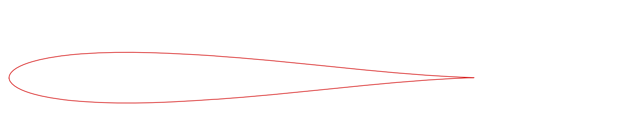


Figure 3: Base profile generated by conformal transformation

MEEN 646 – Aerothermodynamics of Turbomachinery

Module II:

Design of Subsonic Turbine Blade M2-T

& Compressor Blade M2-C

Prepared by:

Dung Tran

Jitaditya Mondal

Submission Date: 02/17/2016

Contents

List of Figures

I. Introduction

The module provides an introduction about axial turbine and axial compressor blades design. For the turbine blade, a Bezier function is utilized for a camberline construction, while in compressor blade, the author uses NACA Camberline equation. A given set of data of base profile is used to plot the suction and pressure lines. Matlab code for plotting those blades are presented in the Appendix.

II. Design Details

A. Nomenclature

V_1	Inlet velocity
V_2	Outlet velocity
α_1	Inlet angle
α_2	Outlet angle
ϕ	Auxiliary angle
γ	Stagger angle
c	Chord length
v	Camber line tangent angle
x_c, y_c	Camber line coordinates
x_p, y_p	Pressure side coordinates
x_s, y_s	Suction side coordinates
C_L^*	Lift coefficient

B. Turbine Blade Design

B.1. Camberline equation

a. Stagger angle based on $\alpha 1$, $\alpha 2$, C – blade chord

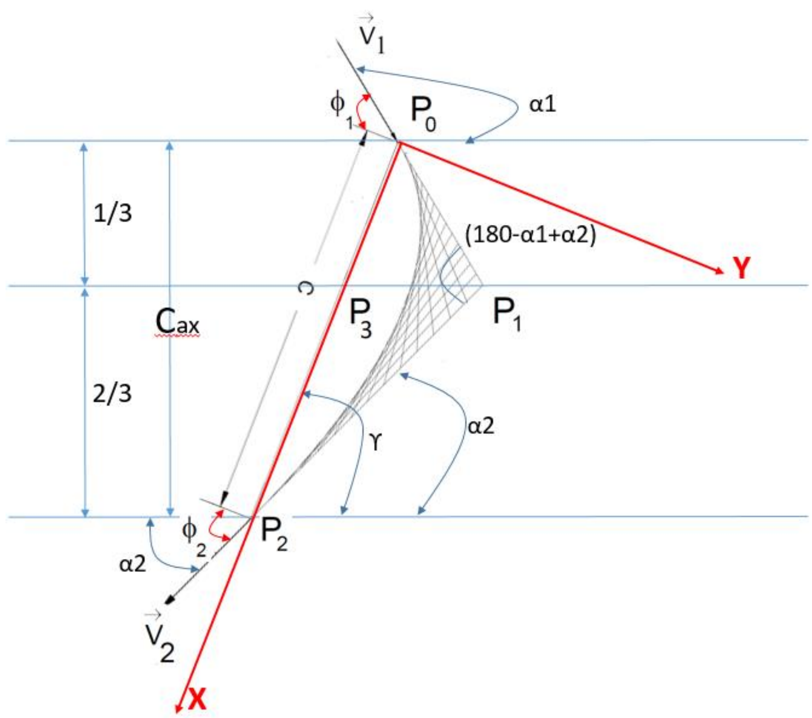


Figure 1: Parameters of the blade

Consider the triangle $\Delta P_0P_1P_2$

$$\textit{Calculate stagger angle given inlet and outlet flow angles}$$

$$\Delta P0P1P2$$

$$\frac{P0P2}{sin(180-\alpha 1+\alpha 2)} = \frac{P0P1}{sin(\Upsilon - \alpha 2)}$$

$$where:\ P0P2 = C\ ,\ P0P1 = \frac{1}{3} * C_{ax}/sin(180-\alpha 1)$$

$$\text{->}\ \frac{C_{ax}}{C} = \frac{3 * sin(\alpha 1) * sin(\Upsilon - \alpha 2)}{sin(\alpha 1 - \alpha 2)}$$

$$Additionally\ \ sin(\Upsilon) = \frac{C_{ax}}{C}$$

$$\rightarrow\ tan(\Upsilon) = \frac{sin(\alpha 2)}{-\frac{1}{3} * \frac{sin(\alpha 1 - \alpha 2)}{sin(\alpha 1)} + cos(\alpha 2)}$$

b. *Bezier function*

Determine the coordinates of P0, P1 and P2

- P0(x_p0,y_p0),

  x_p0 = y_p0 = 0

- P1(x_p1,y_p1),

  Method 1:

  Calculate the area of triangle P1P0P3 using

  P1P0 = 1/3*C/sin(alpha1)*sin(gamma)
  P1P3 = 1/3*C/sin(alpha1)*sin(pi-alpha1-gamma)
  P0P3 = 1/3*C

  Then,

  y_p1 = 2*area/(P0P3);
  x_p1 = y_p1/tan(pi-alpha1-gamma)

  Method2:

  $\phi 1 = \alpha 1 + \Upsilon - \pi/2$

  $\phi 2 = \frac{3\pi}{2} - \alpha 1 - \Upsilon$

  Then,

```
y_p1 = C*(cot(ϕ1)/(1+cot(ϕ1)/cot(ϕ2)));
x_p1 = C*(1/(1+cot(ϕ1)/cot(ϕ2)));
```

- P2(x_p2,y_p2)

  x_p2 = C, y_p0 = 0

c. *Camberline equation*

$$x_{cam(i)} = (1 - zeta)^2 * x_{p0} + 2 * (1 - zeta) * zeta * x_{p1} + zeta^2 * x_{p2};$$

$$y_{cam(i)} = (1 - zeta)^2 * y_{p0} + 2 * (1 - zeta) * zeta * y_{p1} + zeta^2 * y_{p2};$$

where zeta = [0..1]

Camberline tangent angle

$$v_{cam(i)} = atan\left(\frac{-2 * (1 - zeta) * y_{p0} + 2 * (1 - 2 * zeta) * y_{p1} + 2 * zeta * y_{p2}}{-2 * (1 - zeta) * x_{p0} + 2 * (1 - 2 * zeta) * x_{p1} + 2 * zeta * x_{p2}}\right);$$

B.2. Suction and pressure side equation

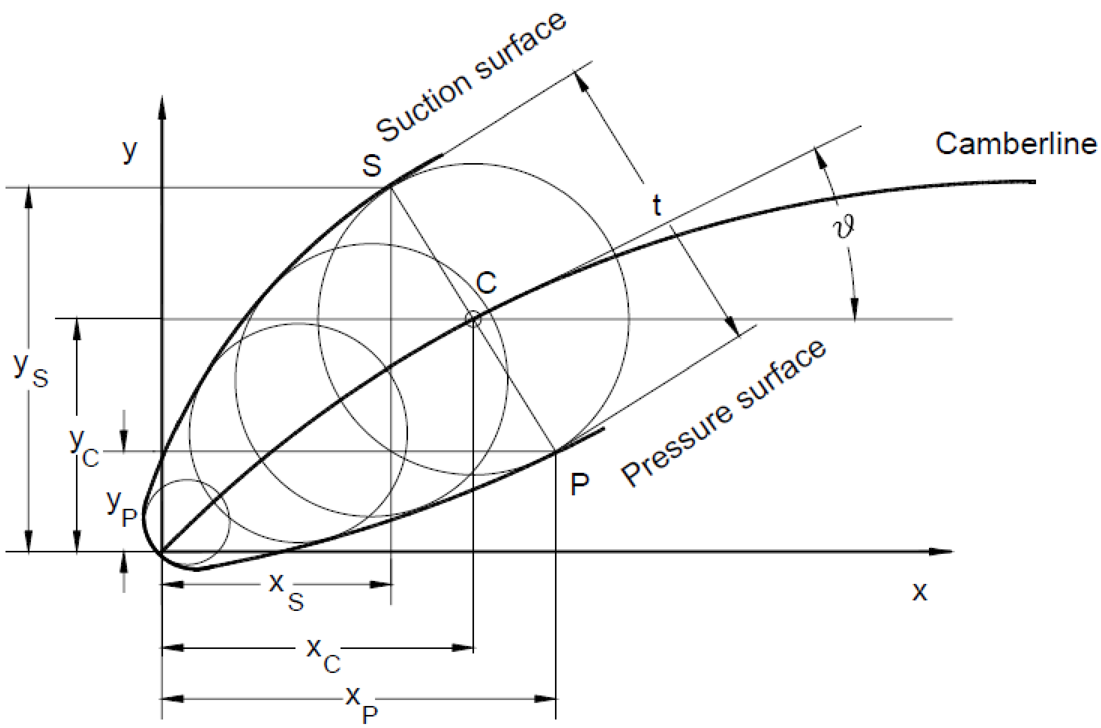


Figure 2: Superimposition of base profile on camberline

A given base profile was superimposed on the camber line to obtain the profile for the suction side and pressure side. The equation used for the suction side was:

$$x_s = x_c - \left(\frac{t}{2}\right)\sin(v)$$

$$y_s = y_c + \left(\frac{t}{2}\right)\cos(v)$$

The equation for the pressure side was

$$x_p = x_c + \left(\frac{t}{2}\right)\sin(v)$$

$$y_p = y_c - \left(\frac{t}{2}\right)\cos(v)$$

B.3. Base profile (thickness)

Given in Appendix A.2

C. Compressor Blade Design

C.1. Camberline equation

For designing NACA compressor blades, the camber line equation is first derived, which is

$$\frac{y_c(x)}{c} = -\frac{C_L^*}{4 * \pi}[\left(1 - \frac{x_c}{c}\right)\ln\left(1 - \frac{x_c}{c}\right) + \frac{x_c}{c}\ln(\frac{x_c}{c})]$$

And camber line angle is given by

$$\frac{dy(x)}{dx} = \tan(v) = \frac{C_L^*}{4 * \pi}\ln\left(\frac{1 - \frac{x_c}{c}}{\frac{x_c}{c}}\right)$$

C.2. Suction and pressure side equation

The formula is given in section B.2. for both turbine and compressor designs.

C.3. Base profile (thickness)

Given in Appendix A.2

III. Results

A. Turbine Blade design

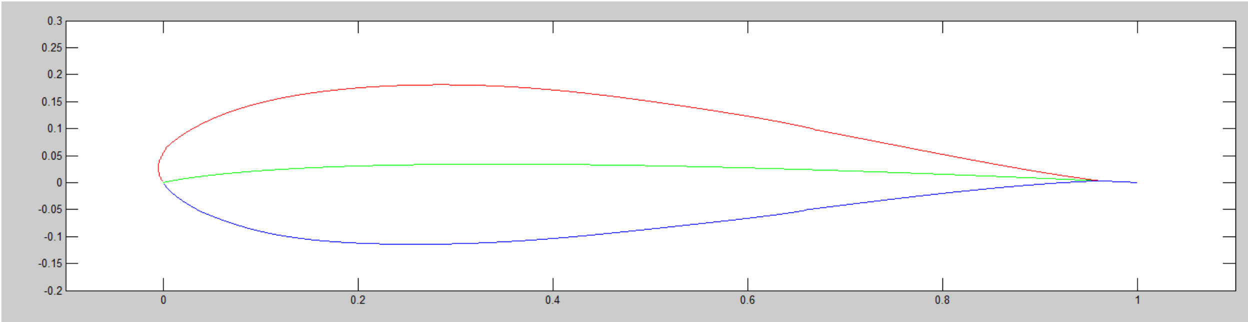


Figure 3: Turbine Blade Design (Single Blade) - Zone 3 (Thick) Baseline Profile, alpha1 = 45deg, alpha2 = 20deg

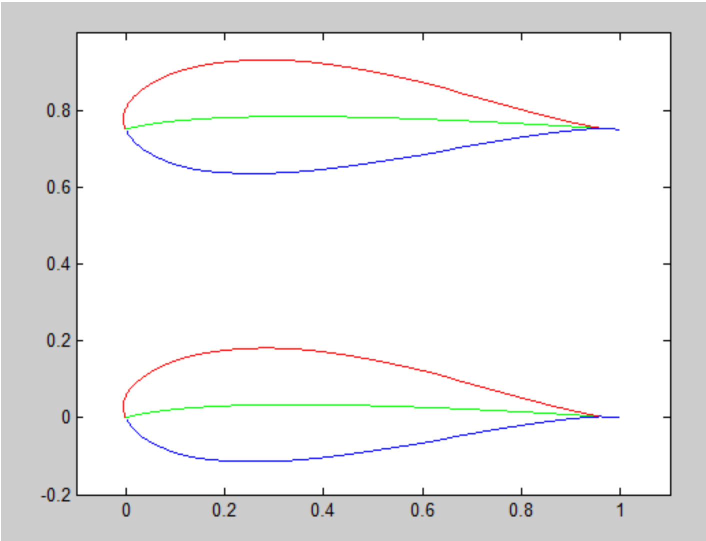


Figure 4: Turbine Blade Design (2 Blades) - Zone 3 (Thick) Baseline Profile, alpha1 = 45deg, alpha2 = 20deg

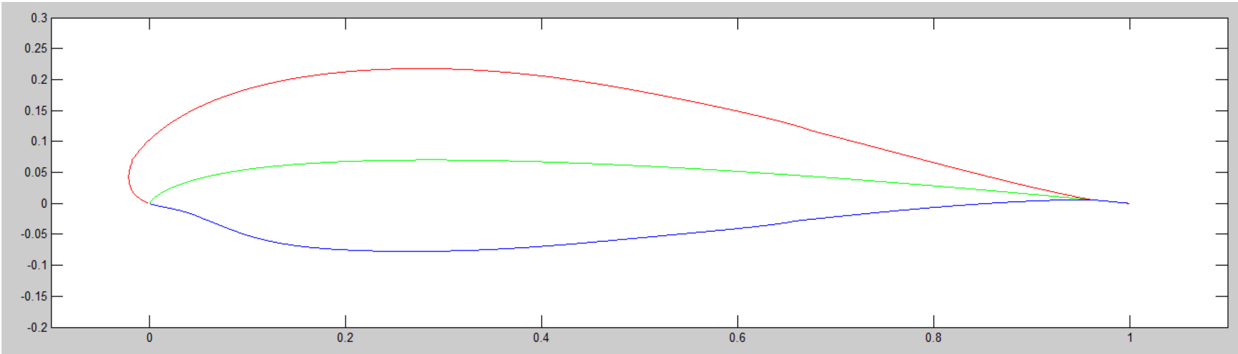


Figure 5: Turbine Blade Design (Single Blade) - Zone 3 (Thick) Baseline Profile, alpha1 = 90deg, alpha2 = 20deg

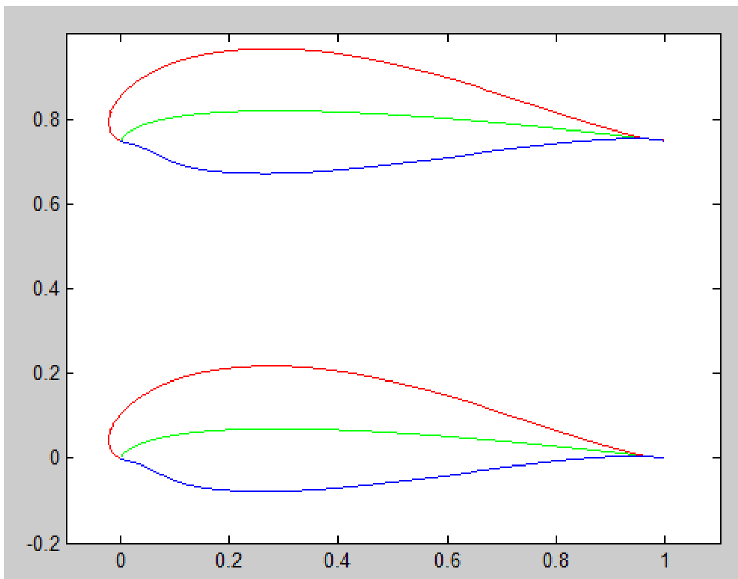


Figure 6: Turbine Blade Design (2 Blades) - Zone 3 (Thick) Baseline Profile, alpha1 = 90deg, alpha2 = 20deg

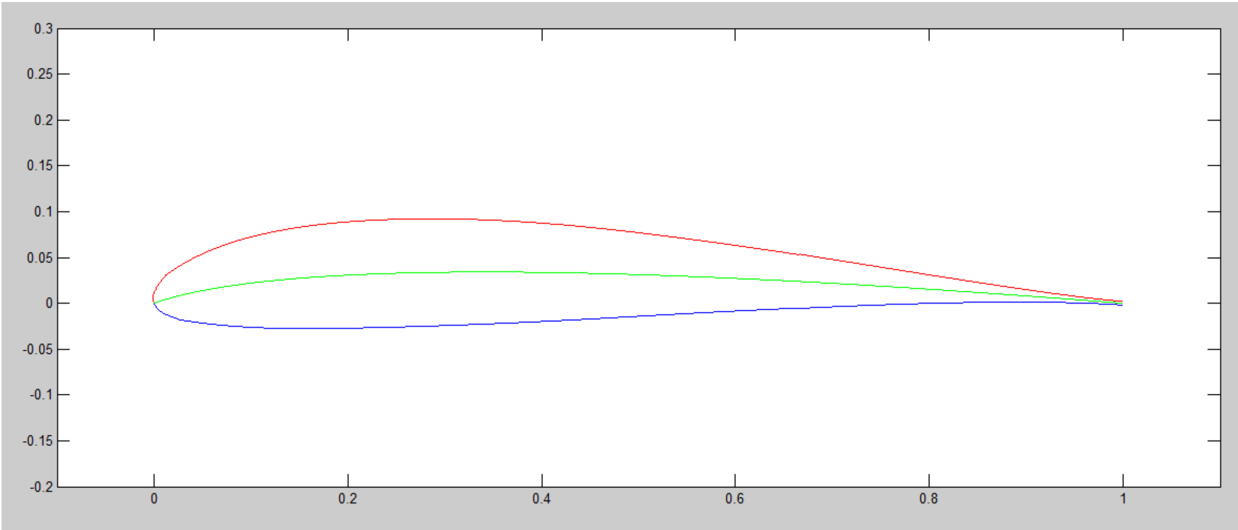


Figure 7: Turbine Blade Design (Single Blade) - Zone 1 (Thin) Baseline Profile, alpha1 = 45deg, alpha2 = 20deg

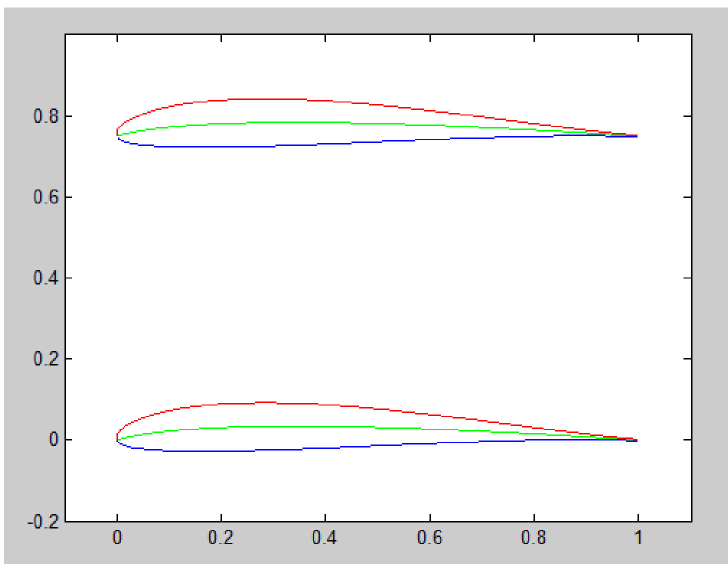


Figure 8: Turbine Blade Design (2 Blades) - Zone 1 (Thin) Baseline Profile, alpha1 = 45deg, alpha2 = 20deg

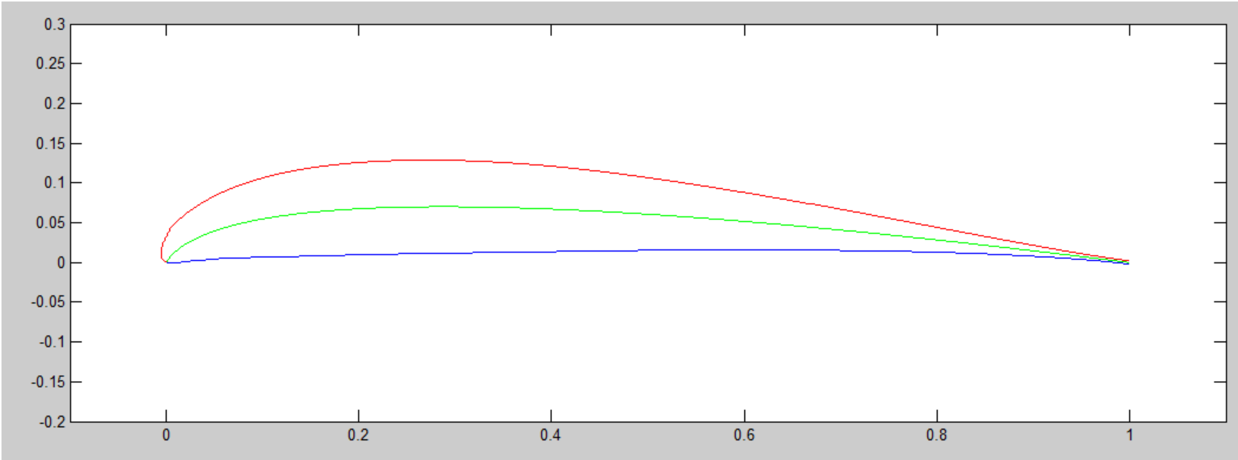


Figure 9: Turbine Blade Design (Single Blade) - Zone 1 (Thin) Baseline Profile, alpha1 = 90deg, alpha2 = 20deg

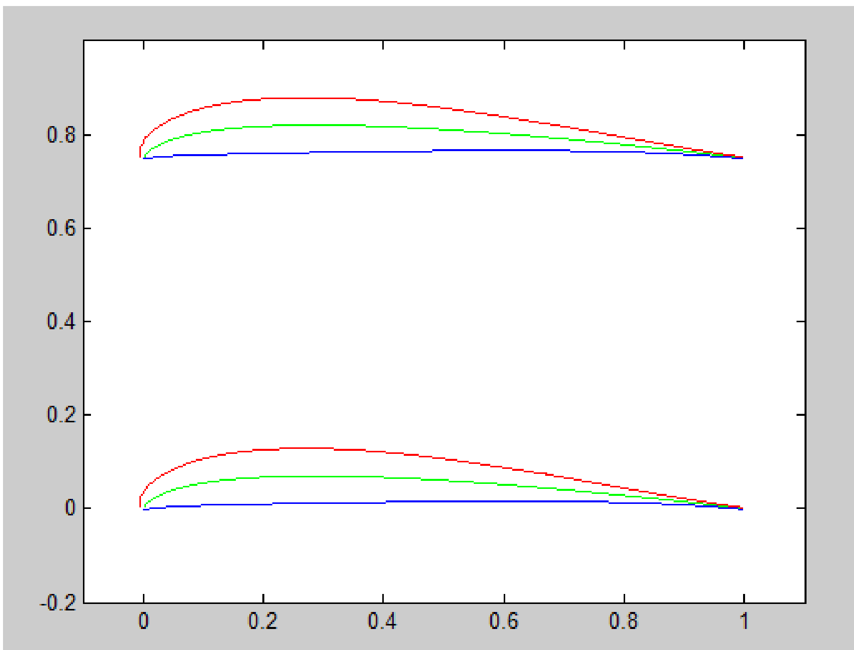


Figure 10: Turbine Blade Design (2 Blades) - Zone 1 (Thin) Baseline Profile, alpha1 = 90deg, alpha2 = 20deg

B. Compressor Blade design

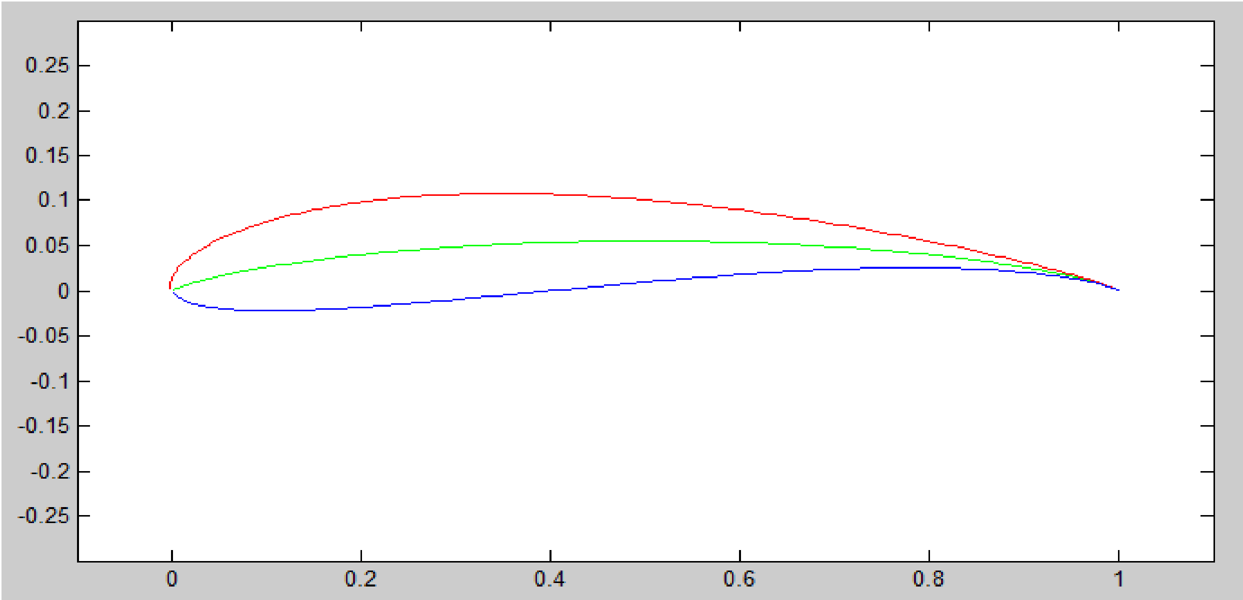


Figure 11: Compressor Blade Design (Single Blade) - Zone 1 (Thin) Baseline Profile, CL=1.0

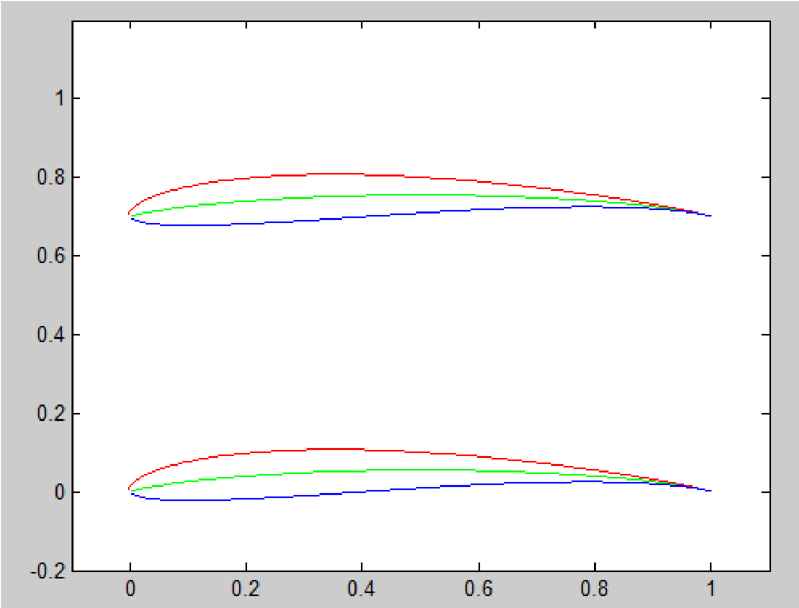


Figure 12: Compressor Blade Design (2 Blades) - Zone 1 (Thin) Baseline Profile, CL=1.0

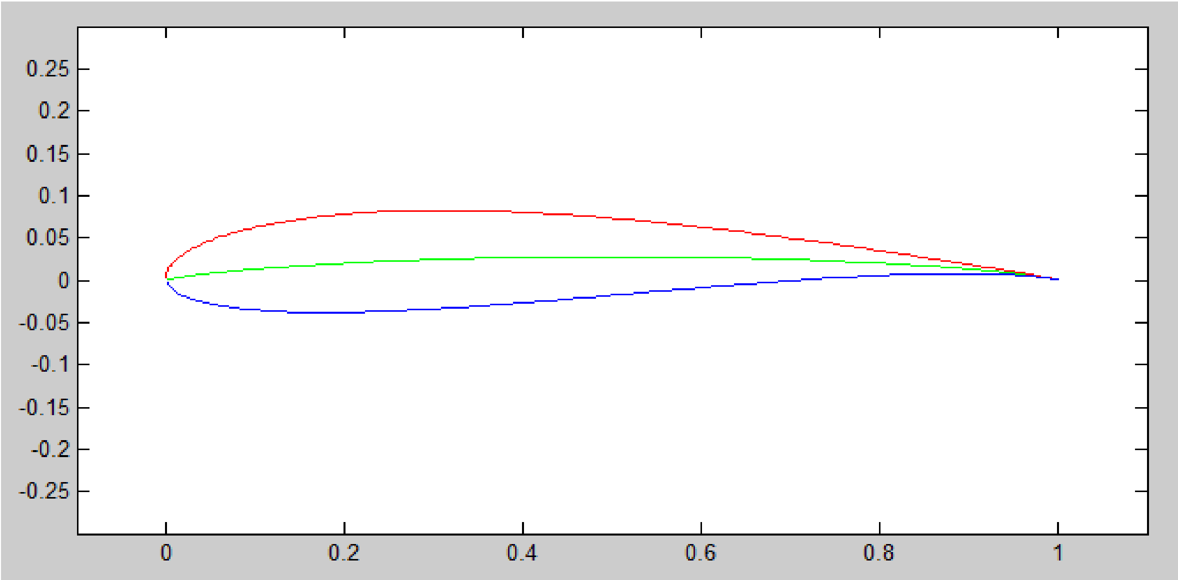


Figure 13: Compressor Blade Design (Single Blade) - Zone 1 (Thin) Baseline Profile, CL=0.5

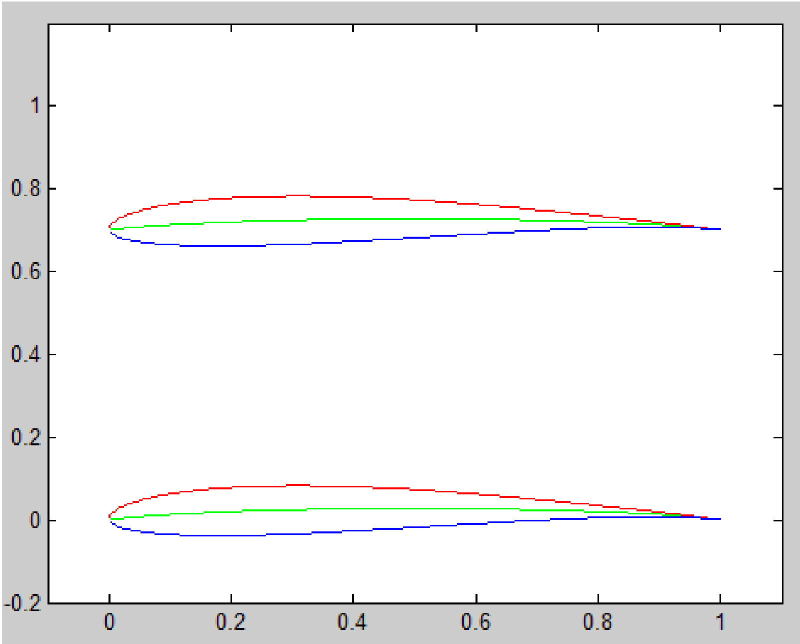


Figure 14: Compressor Blade Design (2 Blades) - Zone 1 (Thin) Baseline Profile, CL=0.5

IV. Discussion and conclusion

- In turbine, increasing deflection, increased the curvature of the camber line.
- In compressor, it was noted that increasing C_L^* increased the curvature of the camber line. Increased curvature can produce more lift

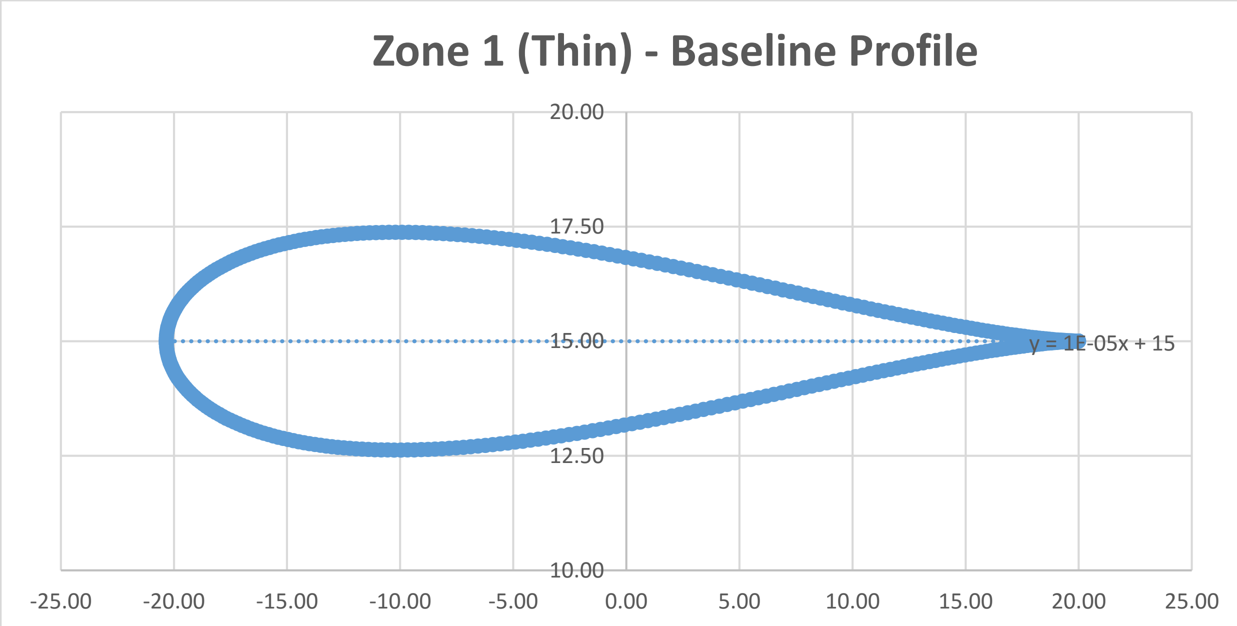
Thus, this module provided an introduction to students in designing of simple blades for turbines and compressors.

REFERENCES

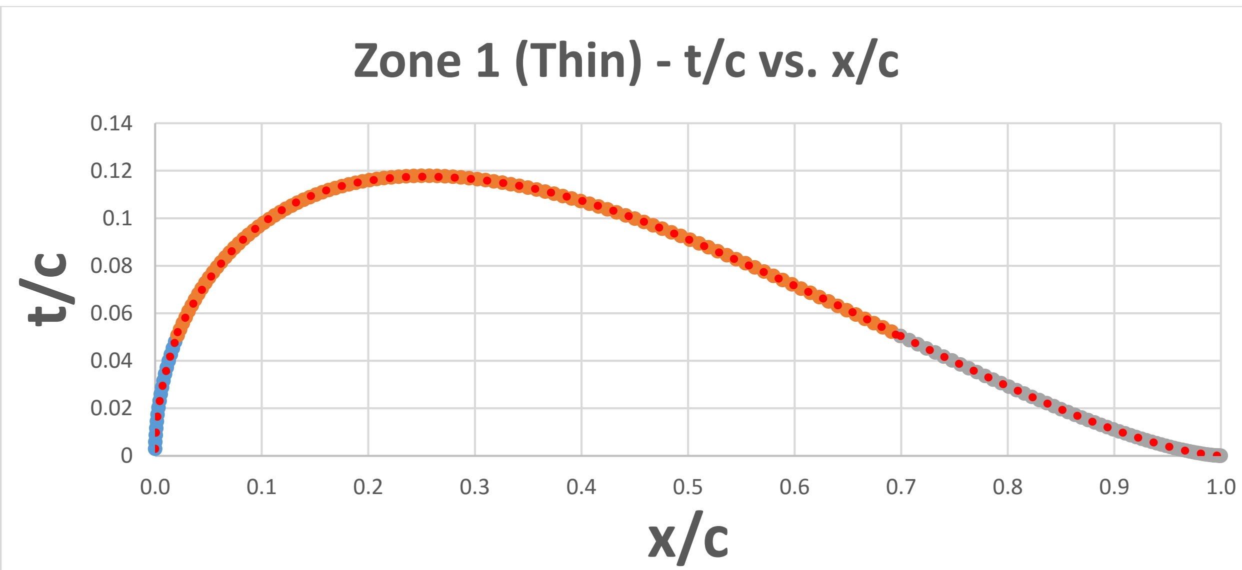
[1] Schobeiri, Meinhard T., *Turbomachinery Flow Physics and Dynamic Performance – Chapter 10*, 2nd Ed. Springer 2012. Print & Online

APPENDIX

A.1 Zone 1 Base-line profile

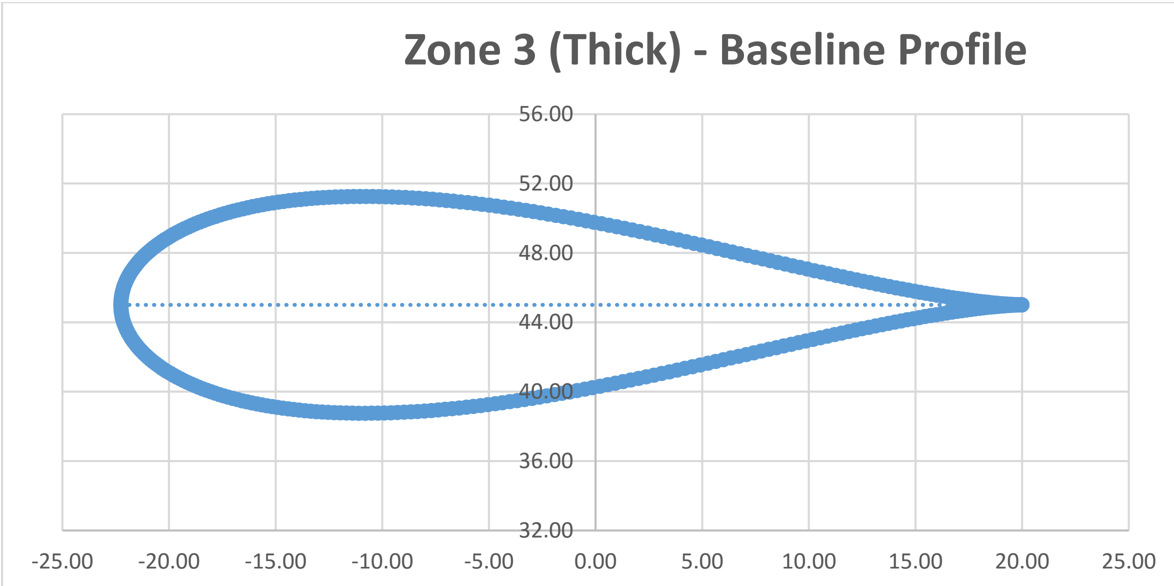


A.2 Zone 1- Baseline profile: t/c vs. x/c

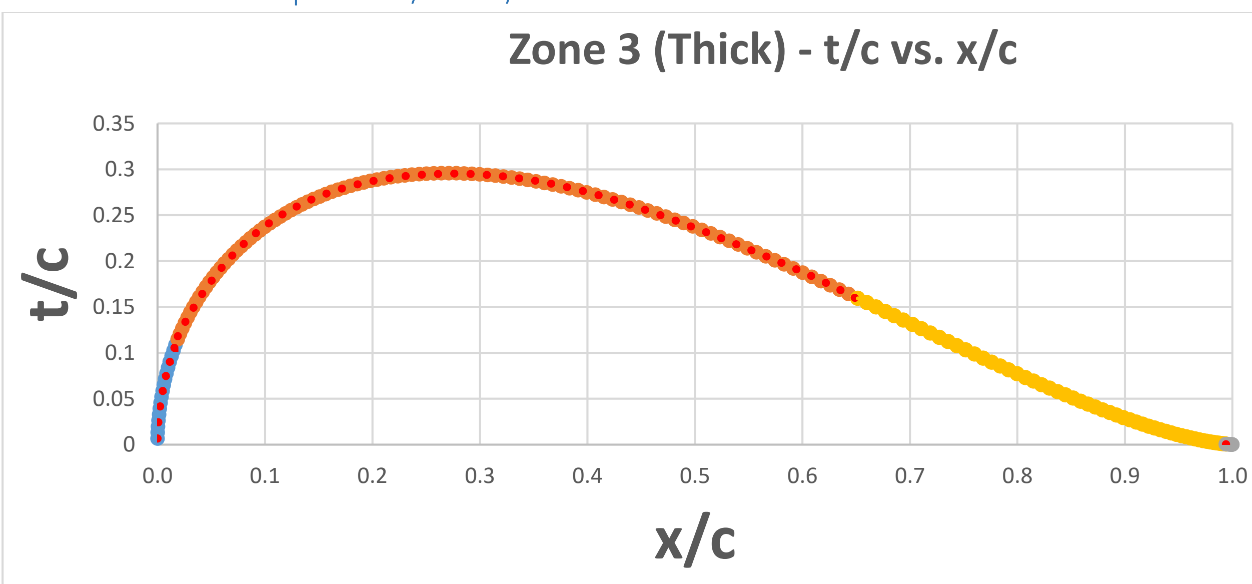


x/c	**t/c as function of x/c**
0 ~ 190/10,000	y = 0.3419x0.4929
191/10,000 ~ 6550/10,000	y = -15.631x6 + 38.563x5 - 38.22x4 + 19.934x3 - 6.2802x2 + 1.1333x + 0.0307
6551/10,000 ~ 1	y = 75.656x6 - 375.15x5 + 774.1x4 - 850.22x3 + 524.07x2 - 172.08x + 23.628

B.1 Zone 3 Base-line profile



B.2 Zone 3- Baseline profile: t/c vs. x/c



x/c	t/c as function of x/c
0 ~ 190/10,000	y = $0.8232x^{0.4941}$
191/10,000 ~ 6550/10,000	y = -56.476x6 + 129.12x5 - 118.24x4 + 56.666x3 - 16.456x2 + 2.8703x + 0.0696
6551/10,000 ~ 9955/10,000	y =65.209x6 - 309.36x5 + 610.82x4 - 641.17x3 + 376.88x2 - 118.1x + 15.713
9956/10,000 ~ 1	y = 7.1679x2 - 14.367x + 7.1992

```
%                 MEEN-646: Module 2
%          Design of Subsonic Compressor Blade M2-C


% Objective:
%    Develop a design software that enables you to generate subsonic compressor blades
%    using NACA Camberline Equation
% Given Parameters
%    - Base profile
%    - Blade chord C
%    - Lift coefficient
%
% Instruction:
%    Input: C, s, CL, iZone selection

clc;

ln = @(x)(log(x));

%input:
iZone=3; % input 1, 2, 3 (1 for thinnest profile and 3 for thickest profile)

%input parameters
CL = 1.0; %lift coefficient
C = 1.0; %blade chord
s = 0.5; %spacing
%n_iter = 10000; %number of iteration

n_iter = 10000;

n_iter_b1 = 190;
n_iter_b2 = 6650;
n_iter_b3 = 9955; %for zone3 only
x_cam1=n_iter_b1/n_iter*C;
x_cam2=n_iter_b2/n_iter*C;
x_cam3=n_iter_b3/n_iter*C;%for zone3 only

%NACA Camberline Equation
for i=1:1:n_iter
    x_cam(i) = i/n_iter*C;
    y_cam(i) = -C*CL/(4*pi)*((1-i/n_iter)*ln(1-i/n_iter)+i/n_iter*ln(i/n_iter));
    %camber line tangent angle
    v_cam(i)=atan(CL/(4*pi)*ln((1-i/n_iter)/(i/n_iter)));

    %for the 2nd camberline
    y_cam1(i) = y_cam(i)+s;
end

%blade thickness
if iZone == 1
    for i=1:1:n_iter_b1
        x(i)=i/n_iter;
        t(i)=C*(0.3419*x(i)^0.4929);%zone1
    end

    for i=(n_iter_b1+1):1:n_iter_b2
        x(i)=i/n_iter;
        t(i)=C*(-15.631*x(i)^6 + 38.563*x(i)^5 - 38.22*x(i)^4 + 19.934*x(i)^3 - 6.2802*x(i)^2 + 1.1333*x(i) + 0.0307);%zone1
    end
```

```
    for i=(n_iter_b2+1):1:n_iter
        x(i)=i/n_iter;
        t(i)=C*(75.656*x(i)^6 - 375.15*x(i)^5 + 774.1*x(i)^4 - 850.22*x(i)^3 + 524.07*x(i)^2 - 172.08*x(i) + 23.628);%zone1
    end

elseif iZone == 2
    for i=1:1:n_iter_b1
        x(i)=i/n_iter;
        t(i)=C*(0.6128*x(i)^0.4937);%zone2
    end

    for i=(n_iter_b1+1):1:n_iter_b2
        x(i)=i/n_iter;
        t(i)=C*(-35.559*x(i)^6 + 83.97*x(i)^5 - 79.529*x(i)^4 + 39.519*x(i)^3 - 11.876*x(i)^2 + 2.0934*x(i) + 0.0531);%zone2
    end

    for i=(n_iter_b2+1):1:n_iter
        x(i)=i/n_iter;
        t(i)=C*(93.702*x(i)^6 - 455.68*x(i)^5 + 921.82*x(i)^4 - 991.96*x(i)^3 + 598.52*x(i)^2 - 192.34*x(i) + 25.931);%zone2
    end

    if t(i)<0
       t(i)=0;
    end

elseif iZone == 3
    for i=1:1:n_iter_b1
        x(i)=i/n_iter;
        t(i)=C*(0.8232*x(i)^0.4941);%zone3
    end

    for i=(n_iter_b1+1):1:n_iter_b2
        x(i)=i/n_iter;
        t(i)=C*(-56.476*x(i)^6 + 129.12*x(i)^5 - 118.24*x(i)^4 + 56.666*x(i)^3 - 16.456*x(i)^2 + 2.8703*x(i) + 0.0696);%zone3
    end

    for i=(n_iter_b2+1):1:n_iter_b3
        x(i)=i/n_iter;
        t(i)=C*(65.209*x(i)^6 - 309.36*x(i)^5 + 610.82*x(i)^4 - 641.17*x(i)^3 + 376.88*x(i)^2 - 118.1*x(i) + 15.713);%zone3
    end

    for i=(n_iter_b2+1):1:n_iter
        x(i)=i/n_iter;
        t(i)=C*(7.1679*x(i)^2 - 14.367*x(i) + 7.1992);%zone3
    end

    if t(i)<0
       t(i)=0;
    end

end


%suction side coordinate
for i=1:1:n_iter
    x_S(i) = x_cam(i) - (t(i)/2)*sin(v_cam(i));
    y_S(i) = y_cam(i) + (t(i)/2)*cos(v_cam(i));
    y_S1(i) = y_S(i) + s;
end
```

```
%pressure side coordinate
for i=1:1:n_iter
    x_P(i) = x_cam(i) + (t(i)/2)*sin(v_cam(i));
    y_P(i) = y_cam(i) - (t(i)/2)*cos(v_cam(i));
    y_P1(i) = y_P(i) + s;
end



%plot the blade
figure(1);
plot(x_cam,y_cam,'g')
hold on
plot(x_S,y_S,'r')
hold on
plot(x_P,y_P,'b')
hold on
%for i=1:1:(n_iter-1)
%    th = 0:pi/50:2*pi;
%    xunit = t(i)/2 * cos(th) + x_cam(i);
%    yunit = t(i)/2 * sin(th) + y_cam(i);
%    h = plot(xunit, yunit,'r');
%    hold on
%end

plot(x_cam,y_cam1,'g')
hold on
plot(x_S,y_S1,'r')
hold on
plot(x_P,y_P1,'b')
hold on
%for i=500:500:(n_iter-500)
%    th = 0:pi/50:2*pi;
%    xunit = t(i)/2 * cos(th) + x_cam(i);
%    yunit = t(i)/2 * sin(th) + y_cam1(i);
%    h = plot(xunit, yunit,'r');
%    hold on
%end

xlim([-.1 1.1])
%ylim([-0.2 0.8])
ylim([-0.2 0.2])
```

```
%                MEEN-646: Module 2
%         Design of Subsonic Turbine Blade M2-T

% Objective:
%    Develop a design software that enables you to generate subsonic turbine blades
%
% Given Parameters
%    - Generate a family of profile (alpha1=90, alpha2=160), (alpha1=45, alpha2=160)
%    - Blade chord C
%
% Instruction:
%    Input: alpha1, alpha2 & iZone

clc;

iZone=1;

%input:
alpha1 = 45; %in degree
alpha2 = 20; %in degree
%convert to rad
alpha1 = (alpha1/180)*pi;
alpha2 = (alpha2/180)*pi;

C = 1.00; %chord
s = 0.75; %spacing
n_iter = 10000; %number of iteration for camberline

n_iter_b1 = 190;
n_iter_b2 = 6650;
n_iter_b3 = 9955; %for zone3 only
x_cam1=n_iter_b1/n_iter*C;
x_cam2=n_iter_b2/n_iter*C;
x_cam3=n_iter_b3/n_iter*C;%for zone3 only

%cascade stagger angle
gamma = atan(sin(alpha2)/(-1/3*sin(alpha1-alpha2)/sin(alpha1)+cos(alpha2)));
C_ax = C*sin(gamma);


%define camber line equation:
x_p0 = 0; y_p0 = 0;
x_p2 = C; y_p2 = 0;

%determine P1 coordinates by consider triangle P0P1P3
%a_1 = 1/3*C/sin(alpha1)*sin(gamma); %P1P0 length
%b_1 = 1/3*C/sin(alpha1)*sin(pi-alpha1-gamma); %P1P3 length
%c_1 = 1/3*C; %P0P3 length
%p = (a_1+b_1+c_1)/2;
%area = sqrt(p*(p-a_1)*(p-b_1)*(p-c_1));

%y_p1 = 2*area/(1/3*C);
%x_p1 = y_p1/tan(pi-alpha1-gamma);

%determine P1 coordinates, formula given in the book (equation 10.40)
phi1=pi/2-alpha1+gamma;
phi2=pi/2+alpha2-gamma;
y_p1 = C*(cot(phi1)/(1+cot(phi1)/cot(phi2)));
x_p1 = C*(1/(1+cot(phi1)/cot(phi2)));
```

```
%Bezier Curve
for i=1:1:n_iter
   zeta(i) = i/n_iter;
   x_cam(i) = (1-zeta(i))^2*x_p0 + 2*(1-zeta(i))*zeta(i)*x_p1 + zeta(i)^2*x_p2;
   y_cam(i) = (1-zeta(i))^2*y_p0 + 2*(1-zeta(i))*zeta(i)*y_p1 + zeta(i)^2*y_p2;

   %camber line tangent angle
   v_cam(i)=atan((-2*(1-zeta(i))*y_p0 + 2*(1-2*zeta(i))*y_p1 + 2*zeta(i)*y_p2)/(-2*(1-zeta(i))*x_p0 + 2*(1-2*zeta(i))*x_p1 + 2*zeta(i)*x_p2));

   %for the 2nd camberline
   y_cam1(i) = y_cam(i)+s;
end

%blade thickness
if iZone == 1
   for i=1:1:n_iter
      x(i)=x_cam(i)/C;
      if (x_cam(i)<x_cam1)
         t(i)=C*(0.3419*x(i)^0.4929);%zone1
      elseif (x_cam(i)>x_cam1) && (x_cam(i)<x_cam2)
         t(i)=C*(-15.631*x(i)^6 + 38.563*x(i)^5 - 38.22*x(i)^4 + 19.934*x(i)^3 - 6.2802*x(i)^2 + 1.1333*x(i) + 0.0307);%zone1
      elseif (x_cam(i)>x_cam2)
         t(i)=C*(75.656*x(i)^6 - 375.15*x(i)^5 + 774.1*x(i)^4 - 850.22*x(i)^3 + 524.07*x(i)^2 - 172.08*x(i) + 23.628);%zone1
      end
   end
elseif iZone == 2
   for i=1:1:n_iter
      x(i)=x_cam(i)/C;
      if (x_cam(i)<x_cam1)
         t(i)=C*(0.6128*x(i)^0.4937);%zone2
      elseif (x_cam(i)>x_cam1) && (x_cam(i)<x_cam2)
         t(i)=C*(-35.559*x(i)^6 + 83.97*x(i)^5 - 79.529*x(i)^4 + 39.519*x(i)^3 - 11.876*x(i)^2 + 2.0934*x(i) + 0.0531);%zone2
      elseif (x_cam(i)>x_cam2)
         t(i)=C*(93.702*x(i)^6 - 455.68*x(i)^5 + 921.82*x(i)^4 - 991.96*x(i)^3 + 598.52*x(i)^2 - 192.34*x(i) + 25.931);%zone2
      end

      if t(i)<0
        t(i)=0;
      end
   end
elseif iZone == 3
   for i=1:1:n_iter
      x(i)=x_cam(i)/C;
      if (x_cam(i)<x_cam1)
         t(i)=C*(0.8232*x(i)^0.4941);%zone3
      elseif (x_cam(i)>x_cam1) && (x_cam(i)<x_cam2)
         t(i)=C*(-56.476*x(i)^6 + 129.12*x(i)^5 - 118.24*x(i)^4 + 56.666*x(i)^3 - 16.456*x(i)^2 + 2.8703*x(i) + 0.0696);%zone3
      elseif (x_cam(i)>x_cam2) && (x_cam(i)<x_cam3)
         t(i)=C*(65.209*x(i)^6 - 309.36*x(i)^5 + 610.82*x(i)^4 - 641.17*x(i)^3 + 376.88*x(i)^2 - 118.1*x(i) + 15.713);%zone3
      elseif (x_cam(i)>x_cam3)
         t(i)=C*(7.1679*x(i)^2 - 14.367*x(i) + 7.1992);%zone3
      end

      if t(i)<0
        t(i)=0;
      end
   end
end
```

```
%suction side coordinate
for i=1:1:n_iter
    x_S(i) = x_cam(i) - (t(i)/2)*sin(v_cam(i));
    y_S(i) = y_cam(i) + (t(i)/2)*cos(v_cam(i));
    y_S1(i) = y_S(i) + s;
end

%pressure side coordinate
for i=1:1:n_iter
    x_P(i) = x_cam(i) + (t(i)/2)*sin(v_cam(i));
    %x_P_test(i) = x_cam(i) - (t(i)/2)*sin(v_cam(i));
    y_P(i) = y_cam(i) - (t(i)/2)*cos(v_cam(i));
    y_P1(i) = y_P(i) + s;
end


%plot the blade
figure(1);
plot(x_cam,y_cam,'g')
hold on
plot(x_S,y_S,'r')
hold on
plot(x_P,y_P,'b')
hold on
%plot(x_P_test,y_P,'y')
%hold on
%for i=1:1000:(n_iter)
%    th = 0:pi/50:2*pi;
%    xunit = t(i)/2 * cos(th) + x_cam(i);
%    yunit = t(i)/2 * sin(th) + y_cam(i);
%    h = plot(xunit, yunit,'r');
%    hold on
%end

plot(x_cam,y_cam1,'g')
hold on
plot(x_S,y_S1,'r')
hold on
plot(x_P,y_P1,'b')
hold on
%plot(x_P_test,y_P1,'y')
%hold on
%for i=1000:1000:(n_iter-1000)
%    th = 0:pi/50:2*pi;
%    xunit = t(i)/2 * cos(th) + x_cam(i);
%    yunit = t(i)/2 * sin(th) + y_cam1(i);
%    h = plot(xunit, yunit,'r');
%    hold on
%end

xlim([-0.1 1.1])
%ylim([-0.2 1])
ylim([-0.2 0.3])
```