MEEN 646 – Aerothermodynamics of Turbomachinery

# Module III:

# Design of a single stage turbine component

*Prepared by:*

Dung Tran

Jitaditya Mondal

Re-Submission Date: 03/24/2016

# Contents



# List of Figures

# I. Introduction

This module provides a preliminary design for a single stage turbine. Using aerodynamic and thermodynamic analysis, the effect of dimensionless coefficients are noted on power and rpm of the turbine. Matlab code is used to plot the variation of these parameters. Using a chosen set of dimensionless coefficients, the stator and rotor blade angles are calculated. Following that, a complete Solidworks design is created of the turbine stage including, nozzle, stator blade, rotor blade, rotor, diffuser, casing, bearing, seal and rotor.

# II. Aero-thermo analysis

The following table shows the provided parameters.

| | |
|---|---|
| $\dot{m}$ | 3.5 kg/s |
| $p_1$ | 1.6 bar |
| $T_1$ | 800 K |
| $\pi$ | 1.2 |
| $\phi$ | 0.4 |
| $r$ | 50% |
| $\eta$ | 0.85 |

The intention of this analysis is to study the effect of mean diameter and axial velocity on the performance of the turbine. Thus, $D_m$ (mean diameter) was varied from 0.2-0.45 m and the inlet axial velocity $V_{ax}$ was varied from 50-100 m/s. Coding was done such that $D_m$ was placed in the outer loop and $V_{ax}$ was placed in the inner loop.

1. Assumption for the calculation:
   - Air to be considered as perfect gas
   - The enthalpy calculation used average number between inlet and outlet condition, e.g. temperature
   - Vax at outlet varies from 50-100 m/s
   - Calculation of power is only based on the axial component
   - Blade heights are constant for both stationary and rotating blades

2. The equations used in the code are:
   a. Stage power

$$P = \dot{m} * l_m = \dot{m}(H_1 - H_2) = \dot{m} * [(h_1 - h_2) - (h_3 - h_2) + \frac{1}{2}(V_1^2 - V_3^2)]$$

   b. Stage flow coefficient

$$\phi = \frac{V_{m3}}{U_3}$$

   c. Stage load coefficient

$$\lambda = \frac{l_m}{U_3^2}$$

d. Flow angles

$$\cot(\alpha_2) = \frac{1}{\phi}\left(1 - r + \frac{\lambda}{2}\right)$$

$$\cot(\alpha_3) = \frac{1}{\phi}\left(1 - r - \frac{\lambda}{2}\right)$$

$$\cot(\beta_2) = \frac{1}{\phi}\left(\frac{\lambda}{2} - r\right)$$

$$\cot(\beta_3) = -\frac{1}{\phi}(\frac{\lambda}{2} + r)$$

$$\lambda = \phi(\cot(\alpha_2) - \cot(\beta_3)) - 1$$

3. Results: On running the code, the following plots were obtained.

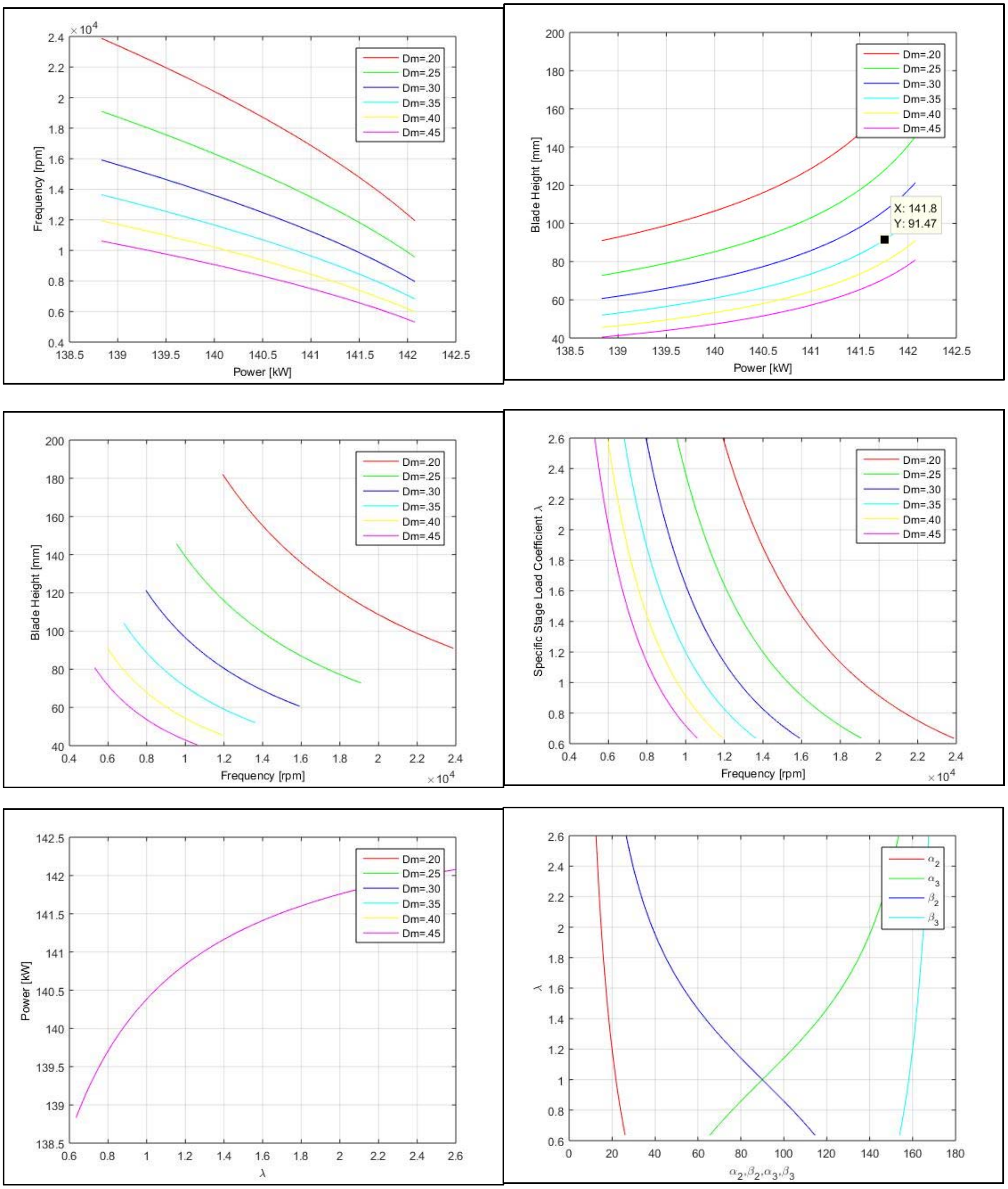


*Figure 1: Relationship between parameters results*

For given value of $D_m$=0.35mm, $\lambda$=2, $c_{stator}$=$c_{rotor}$=40mm, $\sigma$=1.4, the values of the blade angles were found based on the results calculated in the previous section and they have been tabulated below.

| | |
|---|---|
| $\alpha_2$ | 14.91 deg. |
| $\alpha_3$ | 141.5 deg. |
| $\beta_2$ | 38.5 deg. |
| $\beta_3$ | 165.1 deg. |
| Blade height | 91.6 mm |

Velocity diagram:

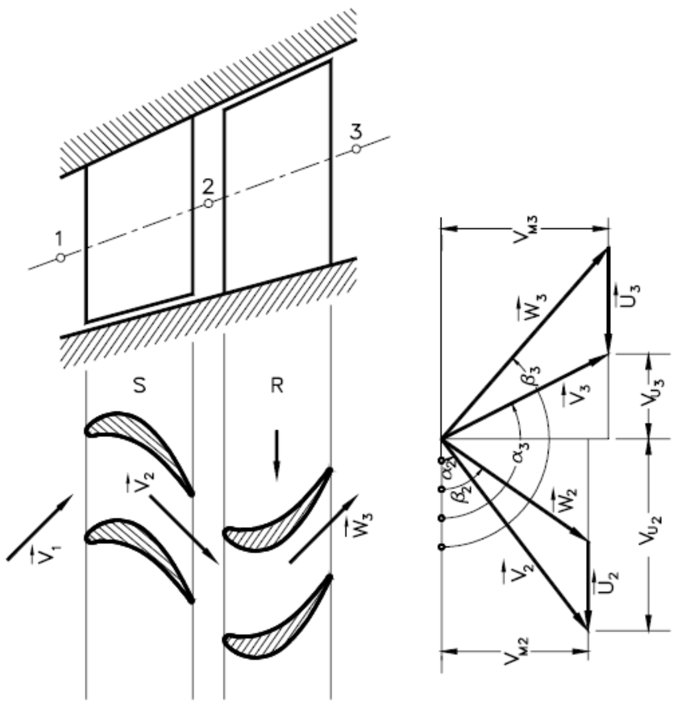


*Figure 2: Velocity Diagram*

## Discussion

From the graphs, we can notice the following trends

- Power can be increased by increasing the mean diameter, blade height and stage load coefficient.
- By increasing the deflection, stage load coefficient can be increased.
- By choosing a specific $\lambda$, a set of blade angles can calculated.

## III. Solidworks Design

1. Blades Design

On inserting these values into the matlab code previously prepared for designing turbine blades, the following plots were obtained. These values were exported to solidworks to produce 3D model of the design. Below are the blade designs along with their cascade.

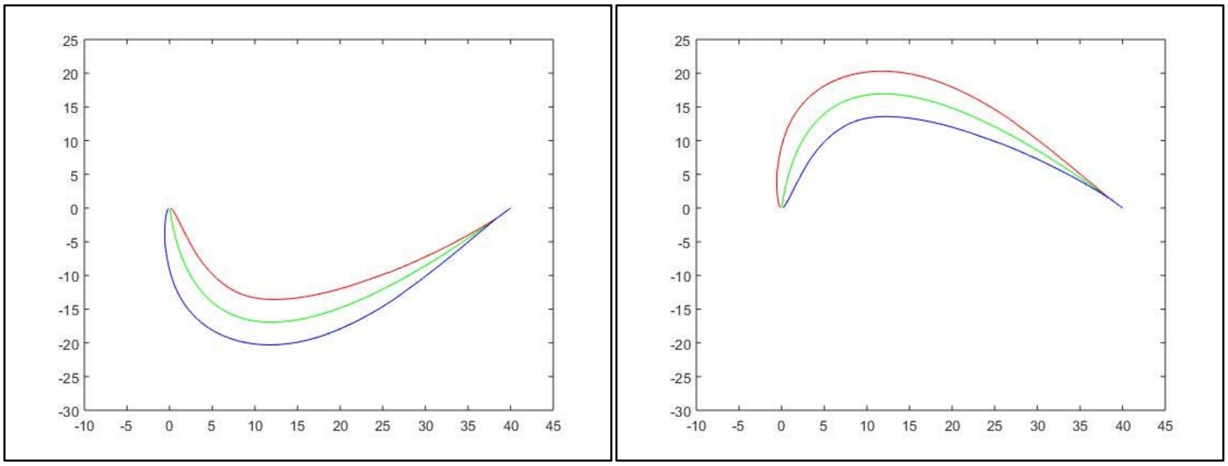

*Figure 3: Stator blade (on the left) and rotor blade (on the right) obtained from Module 2 code*

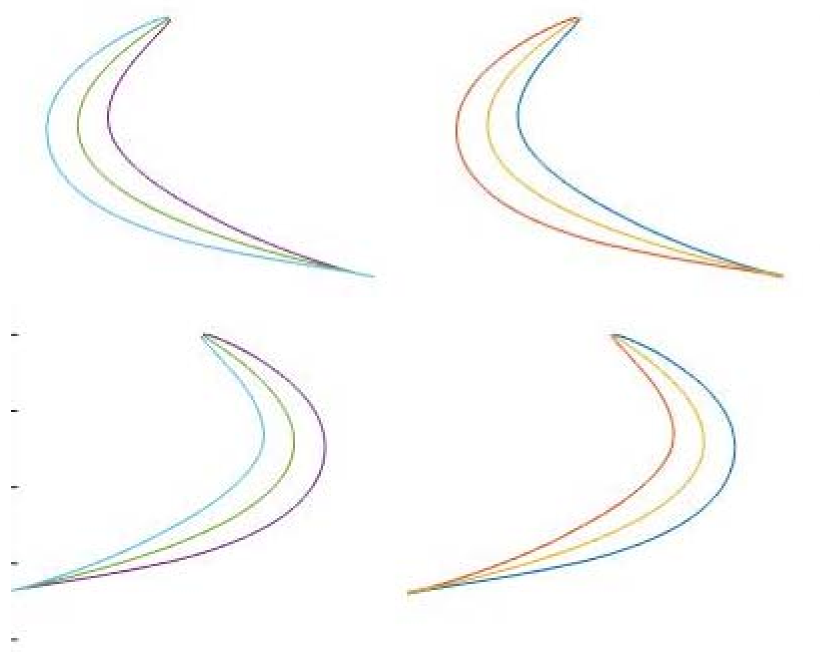

*Figure 4: Blade cascade*

The blades had sharp edges. Radius was added to the trailing edges with radius of 0.2-0.3 mm. Figure below show the turbine rotor blade in Solidworks with trailing edges having radius. Compressor blade has not been provided, but the principle is the same.

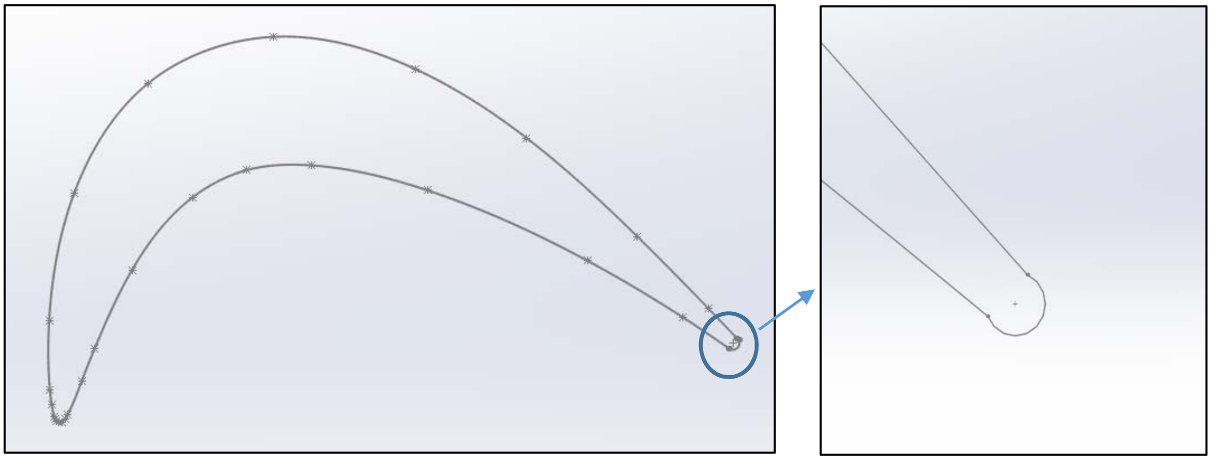

*Figure 5: Solidworks sketch of rotor blade showing radius at trailing edge*

2. Complete Turbine Design

The design of the turbine component was borrowed from "MAN exhaust gas turbocharger TCA series" [2]. The figure below shows the design of the turbocharger.

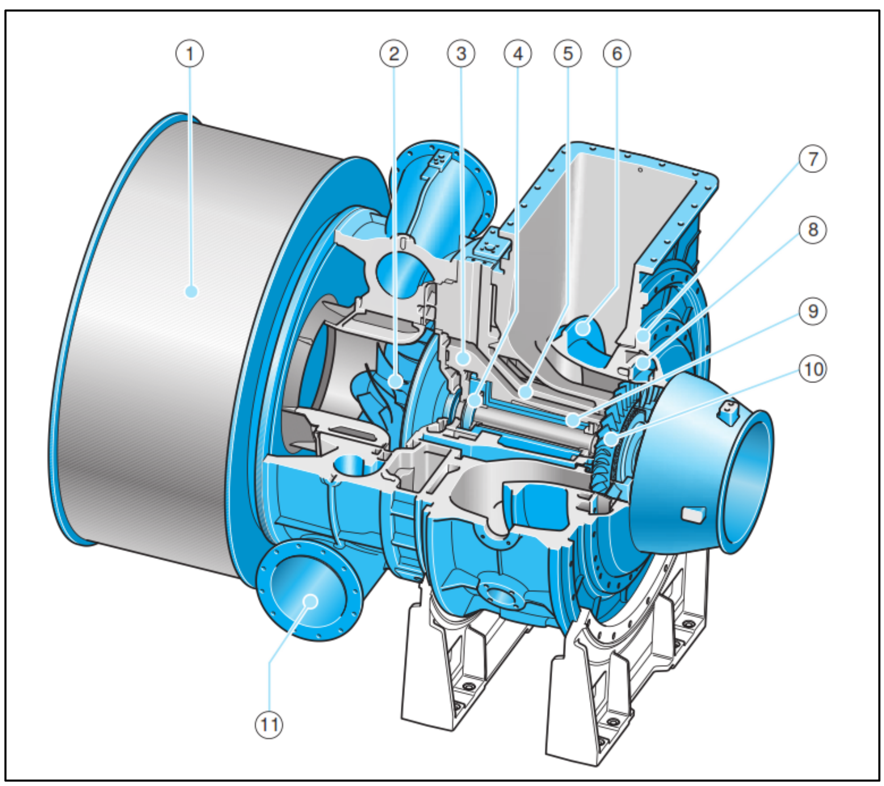


*Figure 6: TCA model design from Man Turbo [2]*

For the preliminary phase of the design of turbine, the following components were considered to be modelled in solidworks:

- Turbine casing
- Inlet nozzle
- Stationary blade and stationary plate
- Rotor blade and rotor wheel
- Diffuser
- Bearing arrangement and bearing housing
- Seal

Detail drawings are attached in the appendix

a. Turbine casing

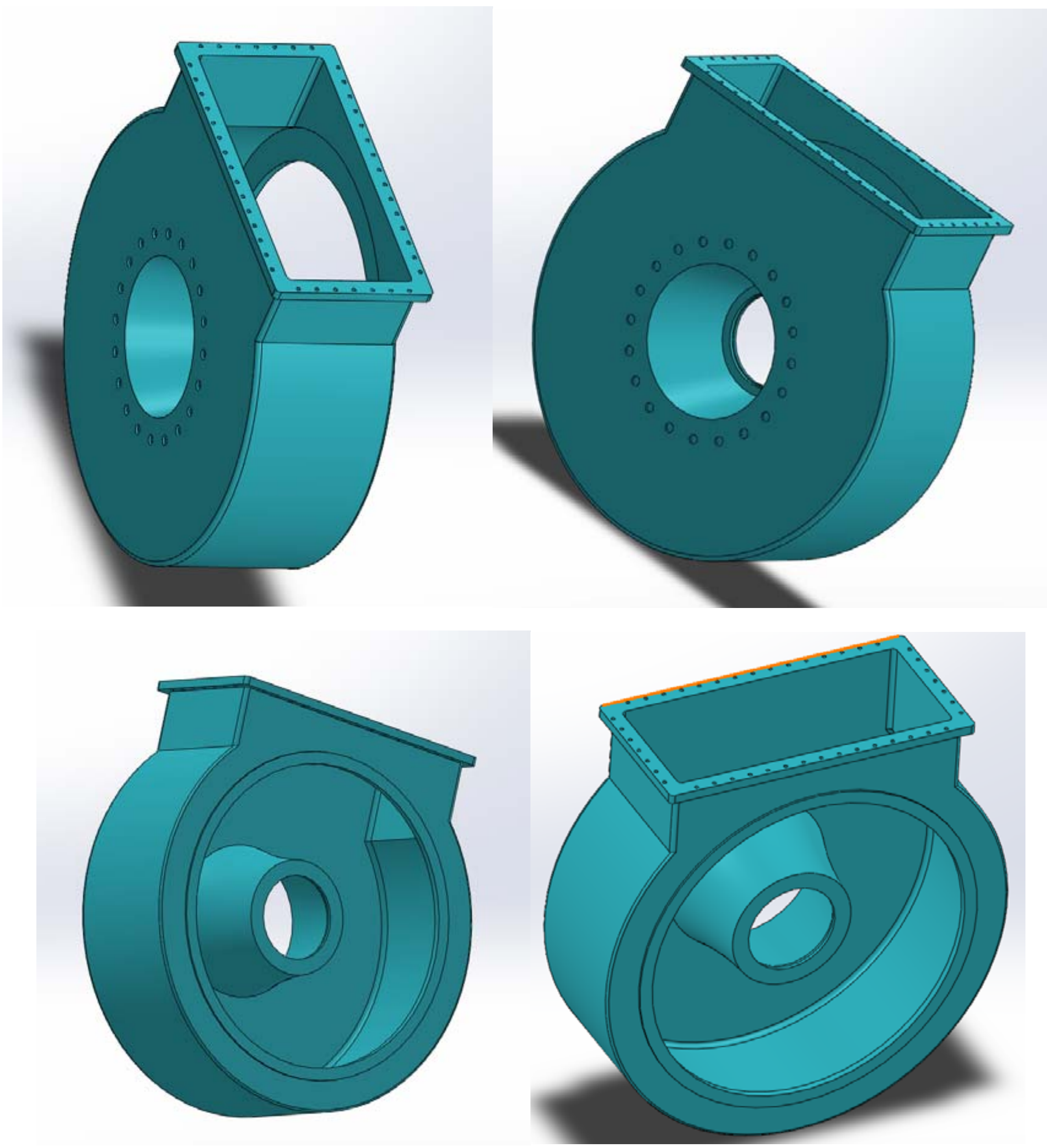

*Figure 7: Turbine casing design*

b. Inlet nozzle

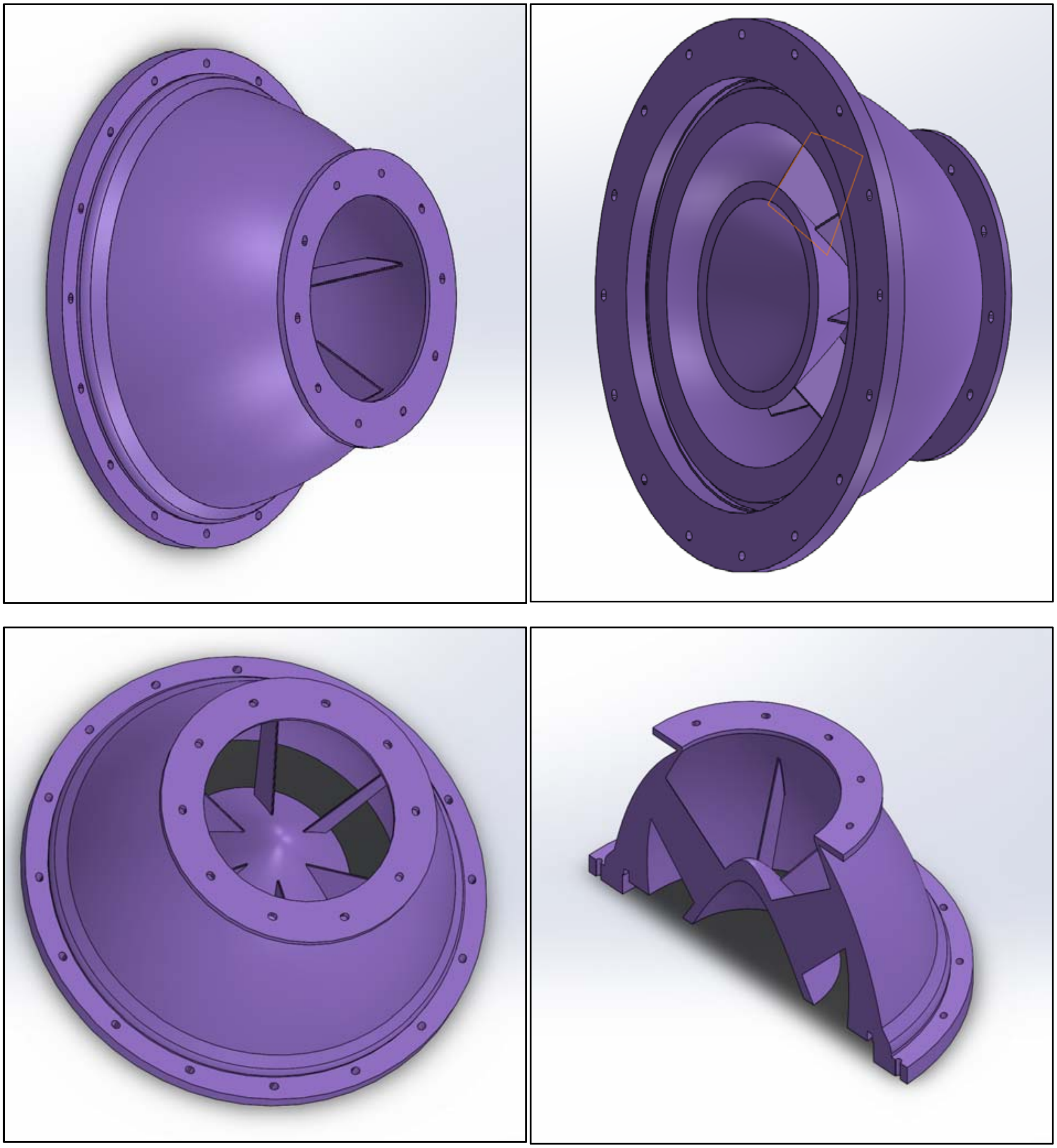

*Figure 8: Inlet nozzle design*

c. Stationary blade and stationary plate

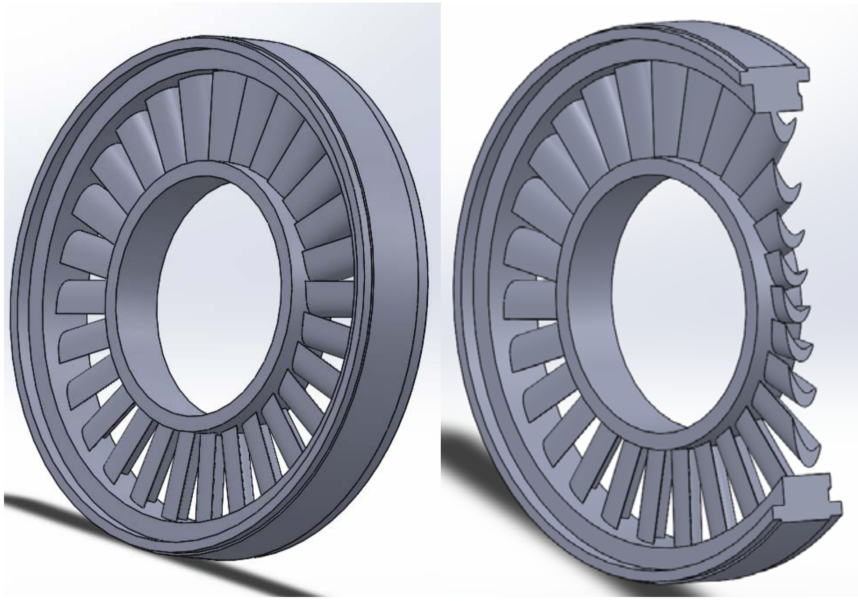

*Figure 9: Stationary blade design*

d. Rotor blade and rotor wheel

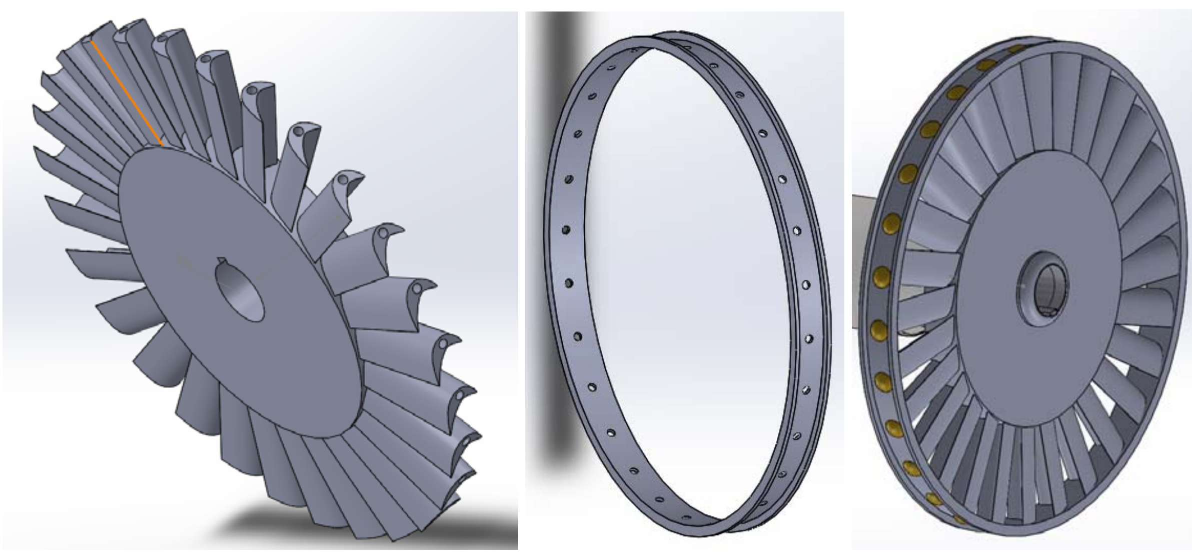

*Figure 10: Rotating blade design*

e. Diffuser

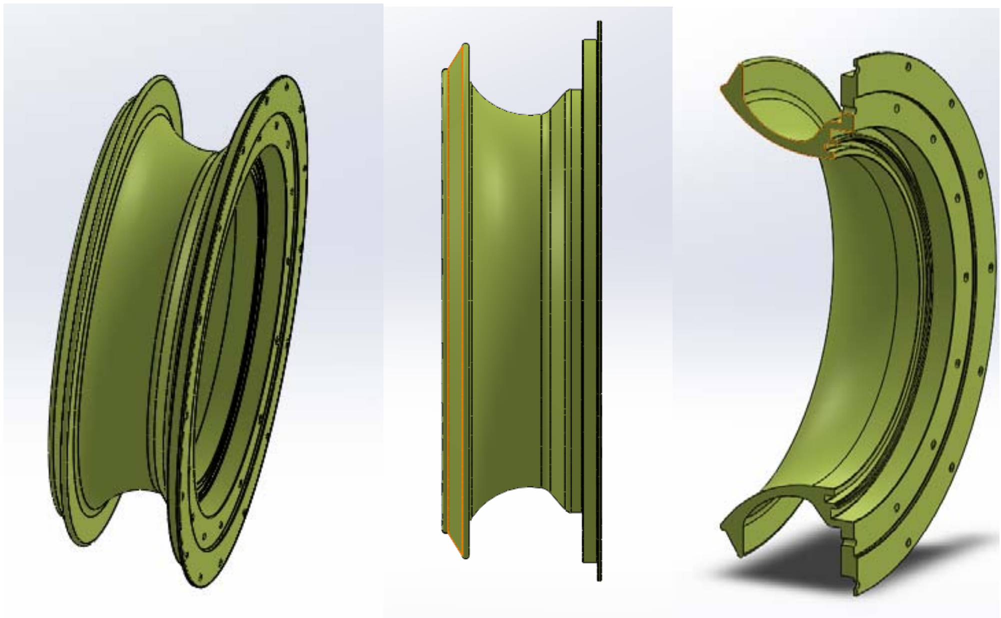

*Figure 11: Diffuser design*

f. Bearing arrangement and bearing housing

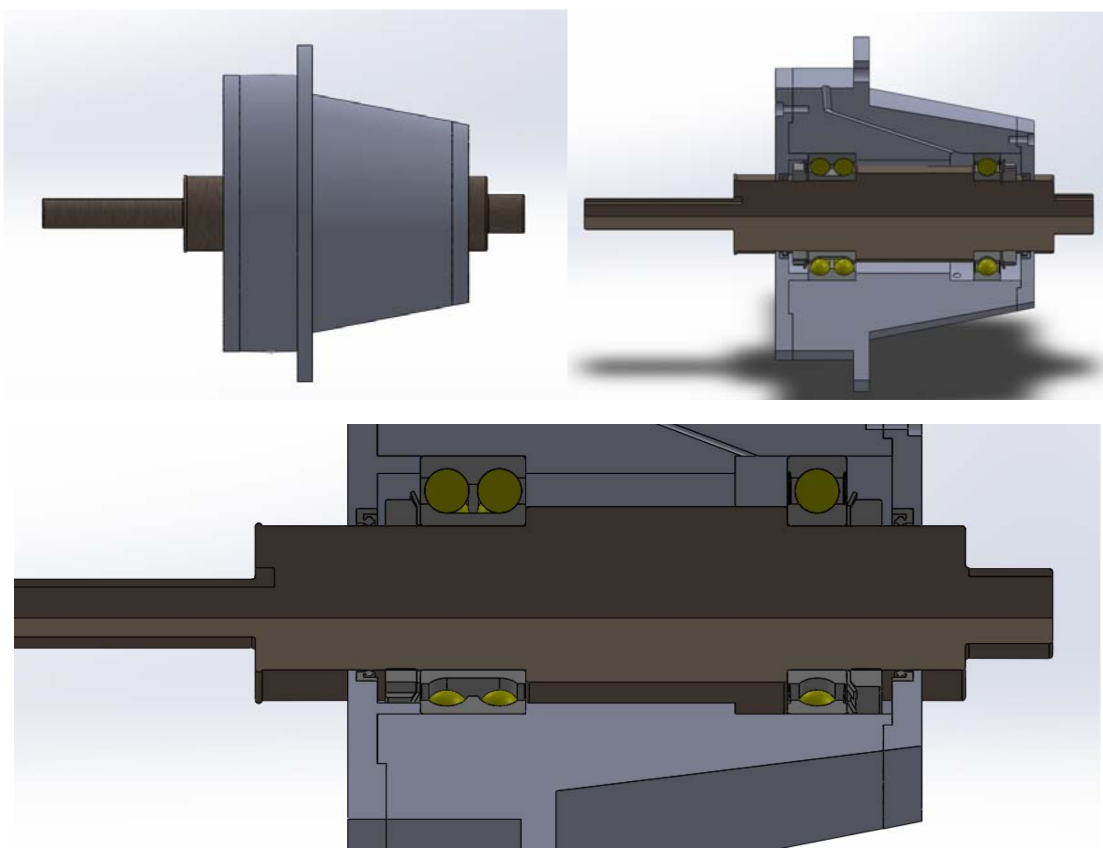

*Figure 12: Bearing arragement*

g. Seal

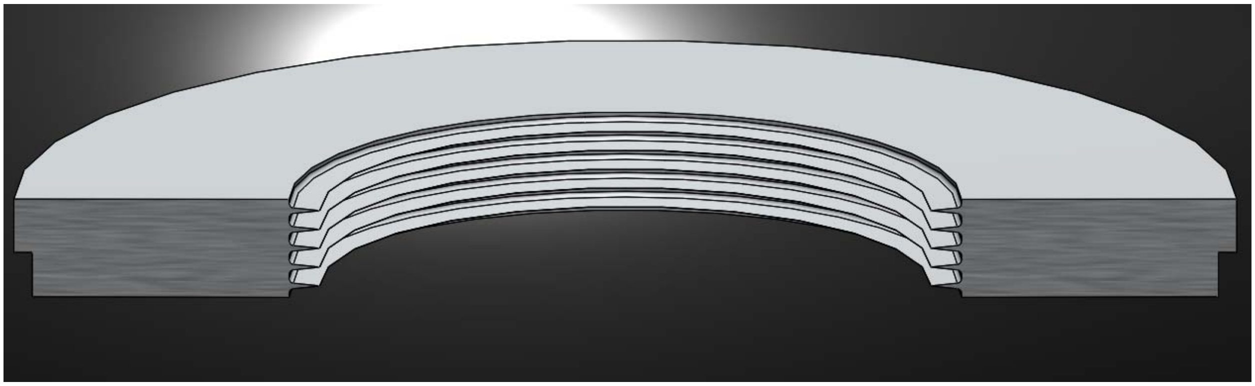

*Figure 13: Gas seal*

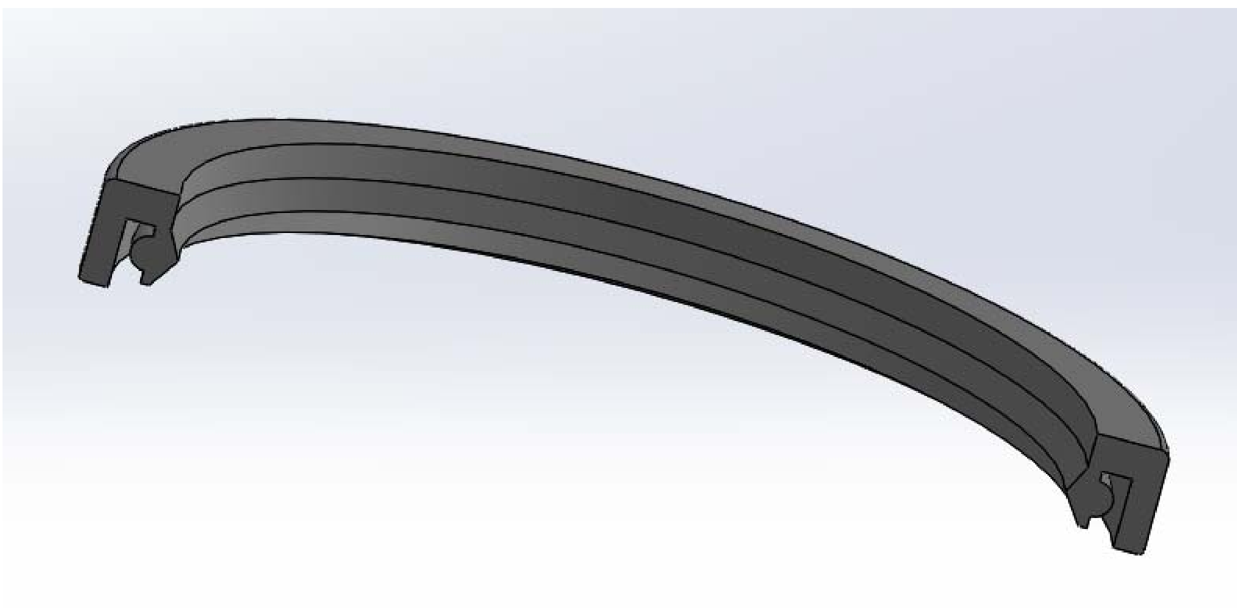

*Figure 14: Oil seal*

e. Full Assembly

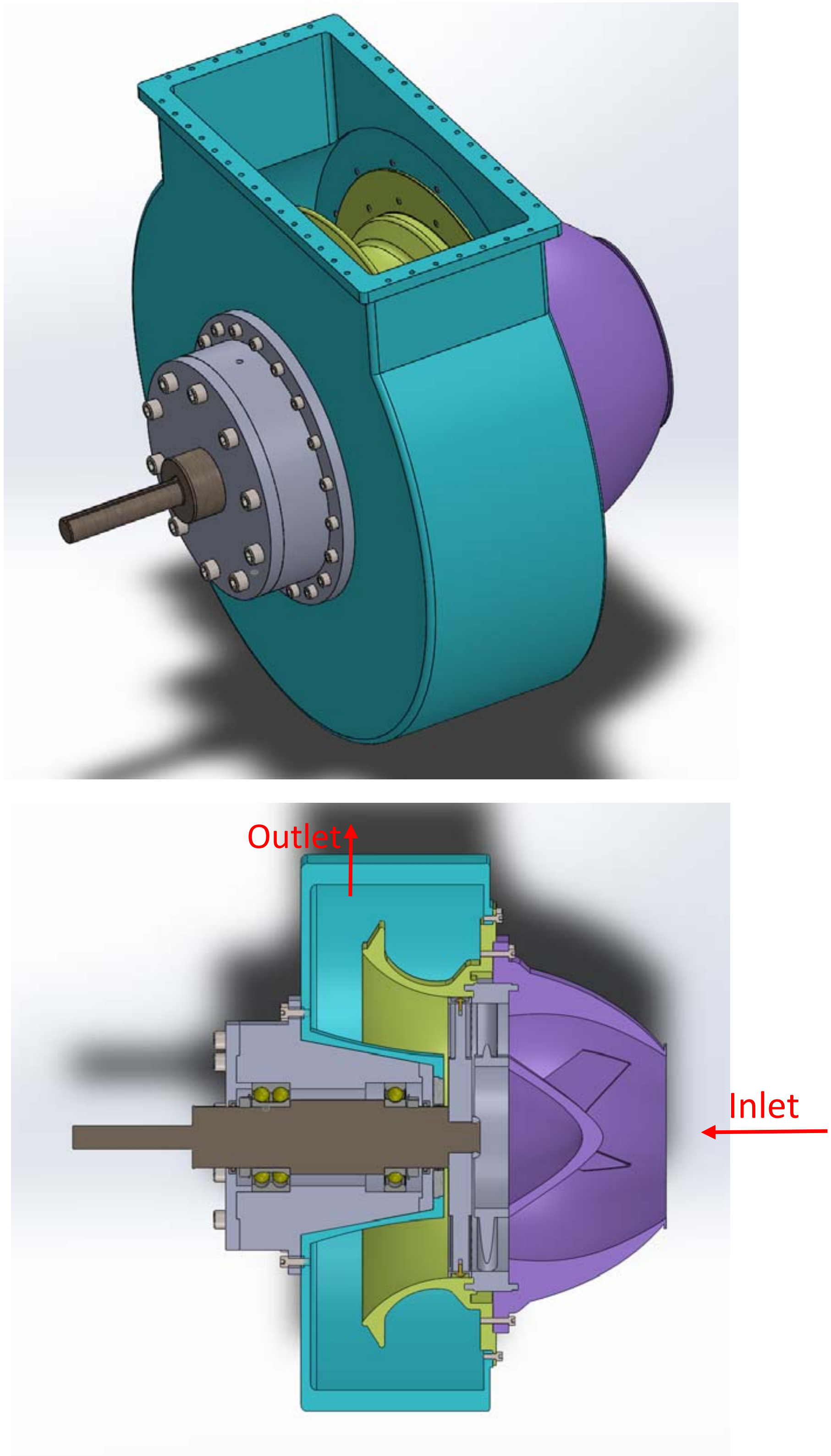


*Figure 15: Full assembly*

f. Explored View

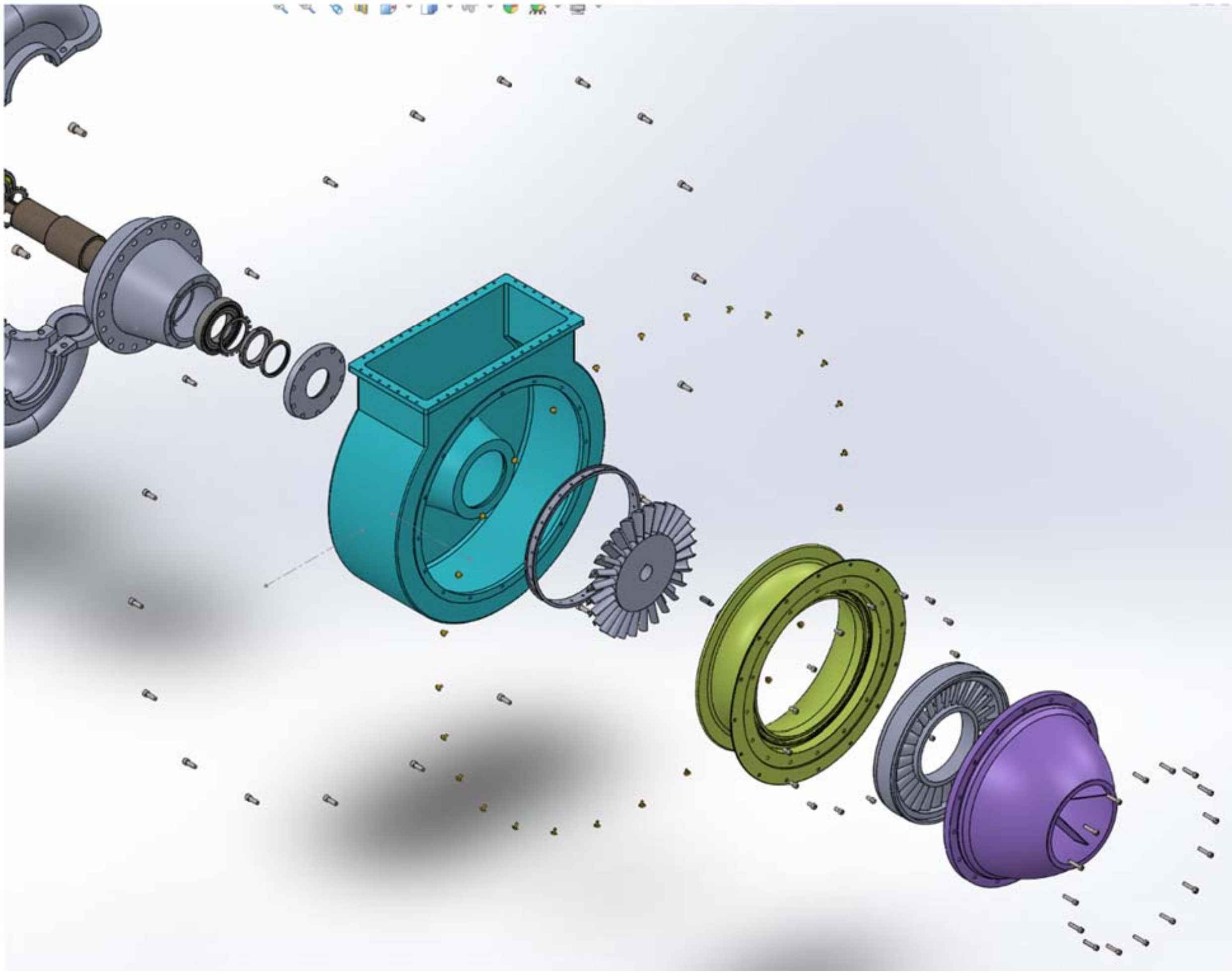

*Figure 16: Exploded view*

## References:

[1] Schobeiri, Meinhard T., *Turbomachinery Flow Physics and Dynamic Performance – Chapter 5*, 2nd Ed. Springer 2012. Print & Online

[2] Online Source: http://turbocharger.man.eu/products/tca

```
%%% MEEN 646 Module 3 %%%

%Initialize constants
% Assumptions:
% (1) Constant diameter and constant cross section
% (2) V1=V1ax, incoming velocity is completely axial
% (3) Perfect gas

clc;


mdot=3.5;
P1=1.6*1000*101.235;
T1=800;
pratio=1.2; %Pressure ratio
phi=0.4; % Stage flow coefficient
r=0.5; % Degree of reaction
eta=0.85; %Isentropic efficiency
Dm=[0.2:0.05:0.45];
V3ax=[50:0.05:100];
k=1.354; % Ratio of specific heat
R=287; % Gas constant
cp1=1098;
cp3=1090;
%alpha1=(90/180)*pi;

%Calculated values
P3=P1/pratio; % Outlet pressure
T3s=T1/((pratio)^((k-1)/k)); %Isentropic temperature
T3=T1-eta*(T1-T3s); %Temperature at exit
rho1=P1/(R*T1); %Density at inlet
rho3=P3/(R*T3); %Density at outlet
h1=cp1*T1;
h3=cp3*T3;

j=1;


for i=1:length(Dm)
    for j=1:length(V3ax)
        V1ax(j)=V3ax(j)*rho3/rho1;
        hb(i,j)=mdot/(rho1*pi*Dm(i)*V1ax(j)); %in m
        hb1(i,j)=mdot/(rho1*pi*Dm(i)*V1ax(j))*1000; %in mm
        Vm3=V3ax(j);
        omega(i,j)=Vm3/(phi*Dm(i)/2); %in rad/s
        omega1(i,j)=omega(i,j)/(2*pi/60); %in rpm
        pw(i,j)=mdot*((h1-h3)+0.5*(V1ax(j)^2-V3ax(j)^2)); %in W
        pw1(i,j)=mdot*((h1-h3)+0.5*(V1ax(j)^2-V3ax(j)^2))/1000; %in kW
        %pw(i,j)=mdot*omega(i,j)*Dm(i)/2*(omega(i,j)*Dm(i)/2+2*tan(alpha1-pi/2)*Vm3)/1000;%in kW
        lambda(i,j)=pw(i,j)/(mdot*(omega(i,j)*Dm(i)/2)^2);
        %lambda(i,j)=1+2*tan(alpha1-pi/2)*Vm3/(omega(i,j)*Dm(i)/2);


        %flow angles (assume constant mean diameter and meridional velocity
        alpha2(i,j)=acot((1/phi)*(lambda(i,j)/2-r+1))*180/(pi); %in degree
        alpha3(i,j)=acot((1/phi)*(-lambda(i,j)/2-r+1))*180/(pi); %in degree
        beta2(i,j)=acot((1/phi)*(lambda(i,j)/2-r))*180/(pi); %in degree
        beta3(i,j)=acot((1/phi)*(-lambda(i,j)/2-r))*180/(pi); %in degree
        if alpha2(i,j) < 0
            alpha2(i,j)=alpha2(i,j)+180;
```

```
        end
        if alpha3(i,j) < 0
           alpha3(i,j)=alpha3(i,j)+180;
        end
        if beta2(i,j) < 0
           beta2(i,j)=beta2(i,j)+180;
        end
        if beta3(i,j) < 0
           beta3(i,j)=beta3(i,j)+180;
        end
        j=j+1;
    end
    i=i+1;
end

figure (1)
plot(pw1(1,:),omega1(1,:),'r')
hold on
plot(pw1(2,:),omega1(2,:),'g')
hold on
plot(pw1(3,:),omega1(3,:),'b')
hold on
plot(pw1(4,:),omega1(4,:),'c')
hold on
plot(pw1(5,:),omega1(5,:),'y')
hold on
plot(pw1(6,:),omega1(6,:),'m')
hold on
xlabel('Power [kW]')
ylabel('Frequency [rpm]')
legend('Dm=.20','Dm=.25','Dm=.30','Dm=.35','Dm=.40','Dm=.45')
grid on

figure (2)
plot(pw1(1,:),hb1(1,:),'r')
hold on
plot(pw1(2,:),hb1(2,:),'g')
hold on
plot(pw1(3,:),hb1(3,:),'b')
hold on
plot(pw1(4,:),hb1(4,:),'c')
hold on
plot(pw1(5,:),hb1(5,:),'y')
hold on
plot(pw1(6,:),hb1(6,:),'m')
hold on
xlabel('Power [kW]')
ylabel('Blade Height [mm]')
legend('Dm=.20','Dm=.25','Dm=.30','Dm=.35','Dm=.40','Dm=.45')
grid on

figure (3)
plot(omega1(1,:),hb1(1,:),'r')
hold on
plot(omega1(2,:),hb1(2,:),'g')
hold on
plot(omega1(3,:),hb1(3,:),'b')
hold on
plot(omega1(4,:),hb1(4,:),'c')
hold on
```

```
plot(omega1(5,:),hb1(5,:),'y')
hold on
plot(omega1(6,:),hb1(6,:),'m')
hold on
xlabel('Frequency [rpm]')
ylabel('Blade Height [mm]')
legend('Dm=.20','Dm=.25','Dm=.30','Dm=.35','Dm=.40','Dm=.45')
grid on

figure (4)
plot(omega1(1,:),lambda(1,:),'r')
hold on
plot(omega1(2,:),lambda(2,:),'g')
hold on
plot(omega1(3,:),lambda(3,:),'b')
hold on
plot(omega1(4,:),lambda(4,:),'c')
hold on
plot(omega1(5,:),lambda(5,:),'y')
hold on
plot(omega1(6,:),lambda(6,:),'m')
hold on
xlabel('Frequency [rpm]')
ylabel('Specific Stage Load Coefficient \lambda')
legend('Dm=.20','Dm=.25','Dm=.30','Dm=.35','Dm=.40','Dm=.45')
grid on

figure (5)
plot(lambda(1,:),pw1(1,:),'r')
hold on
plot(lambda(2,:),pw1(2,:),'g')
hold on
plot(lambda(3,:),pw1(3,:),'b')
hold on
plot(lambda(4,:),pw1(4,:),'c')
hold on
plot(lambda(5,:),pw1(5,:),'y')
hold on
plot(lambda(6,:),pw1(6,:),'m')
hold on
xlabel('\lambda')
ylabel('Power [kW]')
legend('Dm=.20','Dm=.25','Dm=.30','Dm=.35','Dm=.40','Dm=.45')
grid on

figure (6)
plot(alpha2(1,:),lambda(1,:),'r')
hold on
plot(alpha3(2,:),lambda(2,:),'g')
hold on
plot(beta2(3,:),lambda(3,:),'b')
hold on
plot(beta3(4,:),lambda(4,:),'c')
hold on
ylabel('\lambda')
xlabel('\alpha_2,\beta_2,\alpha_3,\beta_3')
legend('\alpha_2','\alpha_3','\beta_2','\beta_3')
grid on
```

```
%                   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=3;

%input:

%stator blade
alpha1 = 120; %in degree
alpha2 = 40; %in degree
a=0;

%rotor blade
% alpha1 = 38.5; %in degree
% alpha2 = 165.1; %in degree
% a=1;

%convert to rad
alpha1 = (alpha1/180)*pi;
alpha2 = (alpha2/180)*pi;

Cax_ratio = 0.4;
C = 40; %chord
s = 50; %spacing
n_iter = 1000; %number of iteration for camberline

n_iter_b1 = 19;
n_iter_b2 = 665;
n_iter_b3 = 995; %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)/(-Cax_ratio*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

      if t(i)<0
        t(i)=0;
      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)/3.5)*sin(v_cam(i));
  y_S(i) = y_cam(i) + (t(i)/3.5)*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)/3.5)*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)/3.5)*cos(v_cam(i));
  y_P1(i) = y_P(i) + s;
end

if a==0
ang=gamma+3*pi/2;
else
  ang=gamma+pi/2;
end
for i=1:length(x_S)
G(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_S(i);y_S1(i)]-[0, 0;-cos(ang), sin(ang)]*[x_S(1);y_S1(1)];
E(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_cam(i);y_cam1(i)]-[0, 0;-cos(ang), sin(ang)]*[x_cam(1);y_cam1(1)];
F(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_P(i);y_P1(i)]-[0, 0;-cos(ang), sin(ang)]*[x_P(1);y_P1(1)];

H(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_S(i);y_S(i)];
I(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_cam(i);y_cam(i)];
J(i,1:2)=[sin(ang), cos(ang);-cos(ang), sin(ang)]*[x_P(i);y_P(i)];
i=i+1;
end

figure(2)
plot(G(:,1),G(:,2));
hold on
plot(F(:,1),F(:,2));
plot(E(:,1),E(:,2));
plot(H(:,1),H(:,2));
plot(I(:,1),I(:,2));
plot(J(:,1),J(:,2));
axis([-20 70 -55 35])
%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


gamma=gamma*180/pi;
x_S=x_S'; x_P=x_P';
y_P=y_P'; y_S=y_S';

%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([-4 44])
%ylim([-0.2 1])
ylim([-30 90])
axis([-10 45 -30 25])
```