DTU Wind Energy

Department of Wind Energy

46300 - Wind Turbine Technology and Aerodynamics

Assignment 3 – 08/12/2017:

Structural

Authors and workloads:

Fauri Giulio (s172424) *(50%)*

Ribaudo Stefano (s171701) *(50%)*

Supervisor:

prof. Martin O.L.Hansen

Index

Introduction

Normally a breakdown is caused by an inadequate control system, extreme wind conditions, fatigue cracks or a defective safety system. A very dangerous breakdown may occur if the power to the generator is lost. There is then no braking torque on the rotor which, in the absence of a safety system such as mechanical or aerodynamic emergency brakes, is free to accelerate. Because the aerodynamic forces increase with the square of the rotor speed, the blades will bend more and more in the downwind direction and might end up hitting the tower or flying off due to centrifugal forces.

Fatigue is a very important issue in a wind turbine construction, which is built to run for a minimum of 20 years, and thus performs in the order of 10^9 revolutions. To estimate the loads on a wind turbine throughout its entire lifetime, the loads and hence the stresses in the material must either be computed using an aeroelastic code in a realistic wind field including turbulence or be measured directly on an existing turbine.

Fig.1 – Section of a blade of a wind turbine

In this report we will analyze the structure and the fatigue strength of the rotor of a wind turbine, answering four questions that deal with the following topics:

1. Q#1 → deflection of a cantilever beam with constant properties;
2. Q#2 → static deflection of the blade using the static loads (constant) and pitch angles for three different wind speeds;
3. Q#3 → eigenfrequencies (natural frequencies) and their corresponding deflections (eigenmodes);
4. Q#4 → stability of an airfoil section of the blade and the work done by the aerodynamic loads for one cycle.

1. Q#1

A blade can be modelled as a beam and when the stiffnesses EI and GI_v at different spanwise stations are computed, simple beam theory can be applied to compute the stresses and deflections of the blade. E is the modulus of elasticity, G is the modulus of elasticity for shear and I denote different moments of inertia.

Values of these parameters are necessary to compute the deflection of a blade for a given load or as input to a dynamic simulation using an aeroelastic code.

EI_1	→	Bending stiffness about first principal axis.
EI_2	→	Bending stiffness about second principal axis.
GI_v	→	Torsional stiffness.
X_E	→	The distance of the point of elasticity from the reference point.
X_m	→	The distance of the centre of mass from the reference point.
X_s	→	The distance of the shear centre from the reference point.
β	→	The twist of the airfoil section measured relative to the tip chordline.
v	→	Angle between chordline and first principal axis.
β+v	→	Angle between tip chordline and first principal axis.

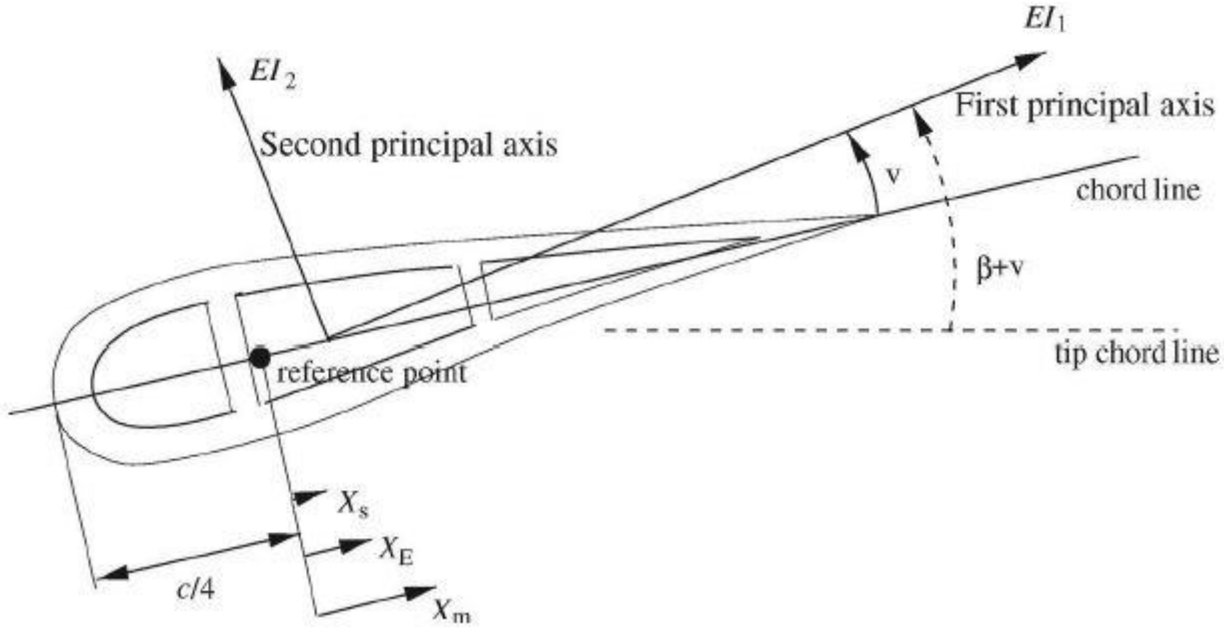


Fig.2 - Section of a blade showing the main structural parameters

Here, we calculate the deflection of a cantilever beam, with constant properties, exposed to a constant external loading. The values of the deflections obtained from the code can be compared with the analytical solution calculated using the formulae given in the assignment.

The loading and deflections are continuously distributed along the beam, but they are calculated at discrete points along the beam using the values of the radius given.

A wind turbine blade can be treated as a technical beam as shown in the following figure (*Fig.3*).

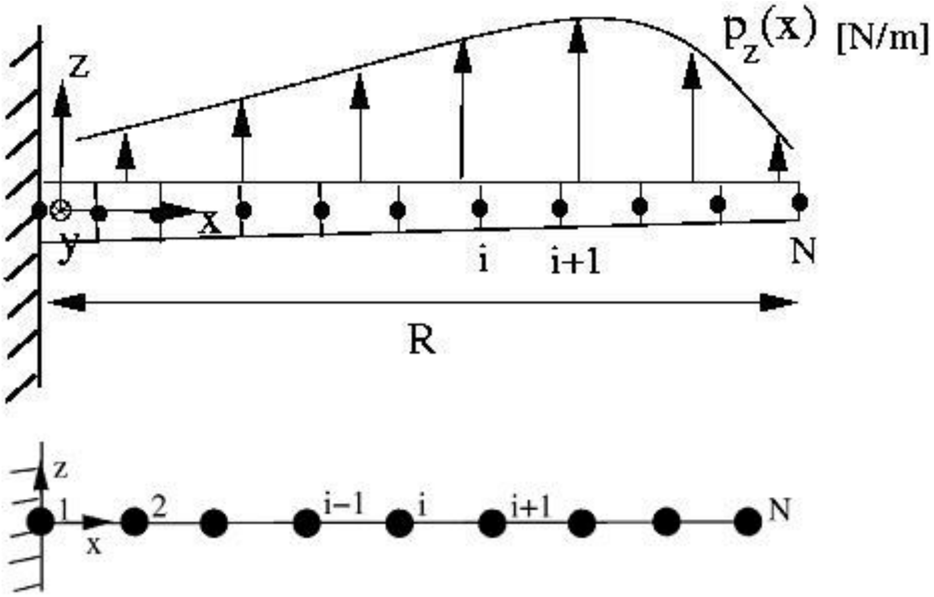
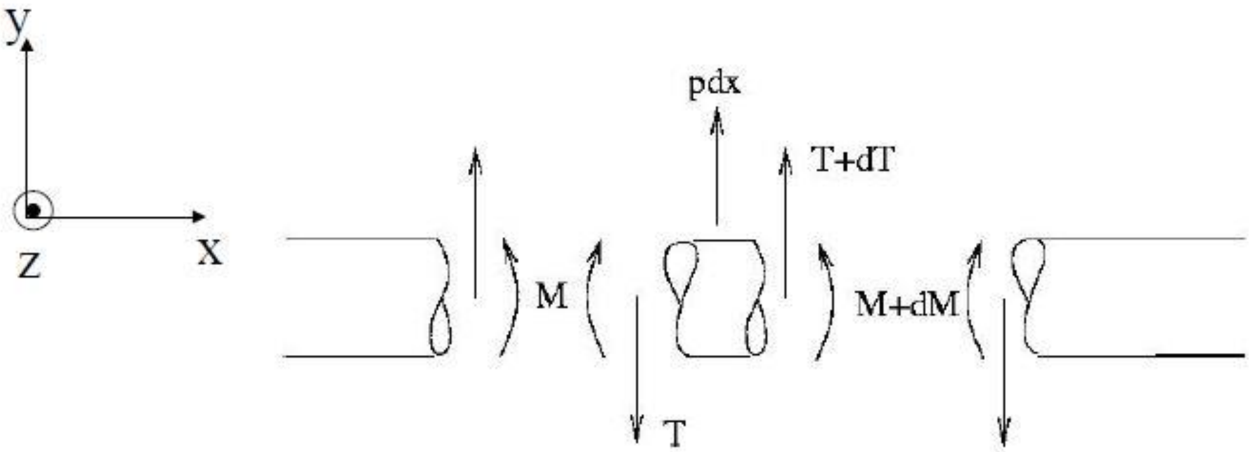


Fig.3 - A wind turbine blade modelled as a cantilever beam

If the external loadings p_y and p_z are known along the blade (we arbitrarily chose a constant value of 2000 [N/m]), the shear forces T_z and T_y and bending moments M_y and M_z can be found as shown in *Fig.4*, applying Newton's second law on an infinitesimal part of the beam.



$$\frac{dT_y}{dx} = -p_y(x) + m(x)\ddot{u}_y(x)$$

$$\frac{dM_z}{dx} = -T_y$$

Fig.4 – Share force and moment of an infinitesimal piece of the beam

In order to obtain correct results, boundary conditions for the cantilever beam need to be considered: the values of shear forces and bending moments are zero at the tip of the beam which correspond to the value calculated at R=30.55m.

With the internal loads (T_y, T_z, M_y, M_z) known, the deflections can be calculated using the Bernoulli beam theory.
First the curvature κ needs to be calculated using the first formulas in *Fig.5*, which is valid only considering principle axes (values of EI1 and EI2 are given as datas in the assignment).

These curvatures are then transformed back to the y-axis and z-axis, then the angular deformations and thus the deflections are calculated, considering the following boundary conditions for the beam: deflections and angles of deflection are zero at the point where the cantilever beam is rigidly clamped to the hub, which we assume to be at r=3.

$$\begin{array}{l} \kappa_1 = \dfrac{M_1}{EI_1} \\ \kappa_2 = \dfrac{M_2}{EI_2} \end{array} \Rightarrow \begin{array}{l} \kappa_z = -\kappa_1 \sin(\beta + v) + \kappa_2 \cos(\beta + v) \\ \kappa_y = \kappa_1 \cos(\beta + v) + \kappa_2 \sin(\beta + v) \end{array} \Rightarrow \frac{d\theta_y}{dx} = \kappa_y$$

Fig.5 – Curvatures, angular deformations and deflections formulas

All the computations and cycles for the first question in our MATLAB code are based on the following figure.

For i=N to 2

$$T_y^{i-1} = T_y^i + \tfrac{1}{2}(p_y^{i-1} + p_y^i)(x^i - x^{i-1})$$
$$T_z^{i-1} = T_z^i + \tfrac{1}{2}(p_z^{i-1} + p_z^i)(x^i - x^{i-1})$$

$$M_y^{i-1} = M_y^i - T_z^i(x^i - x^{i-1}) - (\tfrac{1}{6}p_z^{i-1} + \tfrac{1}{3}p_z^i)(x^i - x^{i-1})^2$$
$$M_z^{i-1} = M_z^i + T_y^i(x^i - x^{i-1}) + (\tfrac{1}{6}p_y^{i-1} + \tfrac{1}{3}p_y^i)(x^i - x^{i-1})^2$$

For i=1 to N

$$M_1^i = M_y^i \cos(\beta^i + v^i) - M_z^i \sin(\beta^i + v^i)$$
$$M_2^i = M_y^i \sin(\beta^i + v^i) + M_z^i \cos(\beta^i + v^i)$$
$$\kappa_1^i = \frac{M_1^i}{EI_1^i}$$
$$\kappa_2^i = \frac{M_2^i}{EI_2^i}$$
$$\kappa_z^i = -\kappa_1^i \sin(\beta^i + v^i) + \kappa_2^i \cos(\beta^i + v^i)$$
$$\kappa_y^i = \kappa_1^i \cos(\beta^i + v^i) + \kappa_2^i \sin(\beta^i + v^i)$$

For i=1,N-1

$$\theta_y^{i+1} = \theta_y^i + \tfrac{1}{2}(\kappa_y^{i+1} + \kappa_y^i)(x^{i+1} - x^i)$$
$$\theta_z^{i+1} = \theta_z^i + \tfrac{1}{2}(\kappa_z^{i+1} + \kappa_z^i)(x^{i+1} - x^i)$$

$$u_y^{i+1} = u_y^i + \theta_z^i(x^{i+1} - x^i) + (\tfrac{1}{6}\kappa_z^{i+1} + \tfrac{1}{3}\kappa_z^i)(x^{i+1} - x^i)^2$$
$$u_z^{i+1} = u_z^i - \theta_y^i(x^{i+1} - x^i) - (\tfrac{1}{6}\kappa_y^{i+1} + \tfrac{1}{3}\kappa_y^i)(x^{i+1} - x^i)^2$$

Fig.6 - Formulas used in MATLAB code

In the next plot it is possible to notice that the values of deflection (u_y and u_z) along y and z axes, at every discrete point considered, perfectly match with the results obtained from the analytical solution.

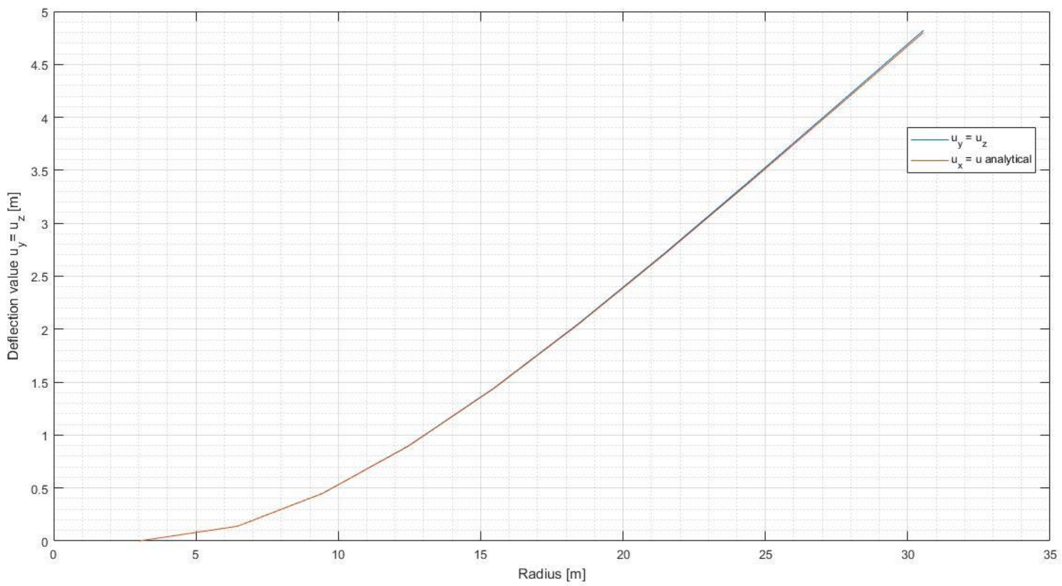


Fig.7 – Deflection along the beam cantilever

The values of u_y and u_z are equal because we chose p_y=p_z. In order to obtain the same results using the two methods we considered that at R=3 the beam cantilever is rigidly clamped to the hub, so the length of the beam is considered as R= r(N) – r(1) = 30.55 – 3 = 27.55 [m].

2. Q#2

In the first question we checked the code against the analytical solution for a beam with constant properties, EI=const, constant loading, p=const and no twist.

Here, we calculate the static deflection of the blade described above using the static loads (constant) and pitch angles in the table given in the assignment (that are in the same order as the aerodynamic loads for the Tjaereborg WT) for the three wind speeds V_0 = (8, 12, 20) m/s.

We assume that the first principal axes are aligned with the chord of the blade sections and we calculate the deflections in a coordinate system that follows the blade tip.

In order to find the distribution of the external loads along these axes, with normal and tangential loads given (p_{tan}, p_{norm}) we need to consider the pitch angles of the blade for every value of V_0, and calculate the projections of p_{tan} and p_{norm} on y-z axes using the following formulae:

$$p_y = p_{tan} * \cos(\theta_p) - p_{norm} * \sin(\theta_p)$$

$$p_z = p_{tan} * \sin(\theta_p) + p_{norm} * \cos(\theta_p)$$

For V_0=8 m/s and V_0=12 m/s the values of the pitch angle (θ_p) are zero, so the y-z axes correspond to the normal and tangential ones, therefore $p_{tan}=p_y$ and $p_{norm}=p_z$.

Figures 8 and 9 represent the deflections obtained, respectively along the y and z axes for every value of the radius given, for the three values of velocities considered.

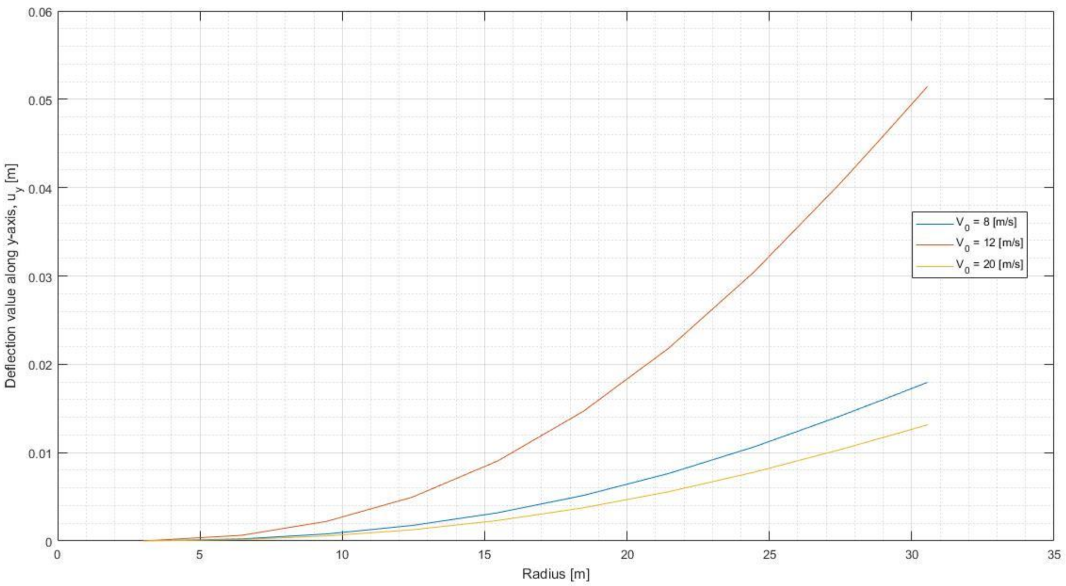


Fig.8 - deflection values and angles along the beam cantilever (u_y)

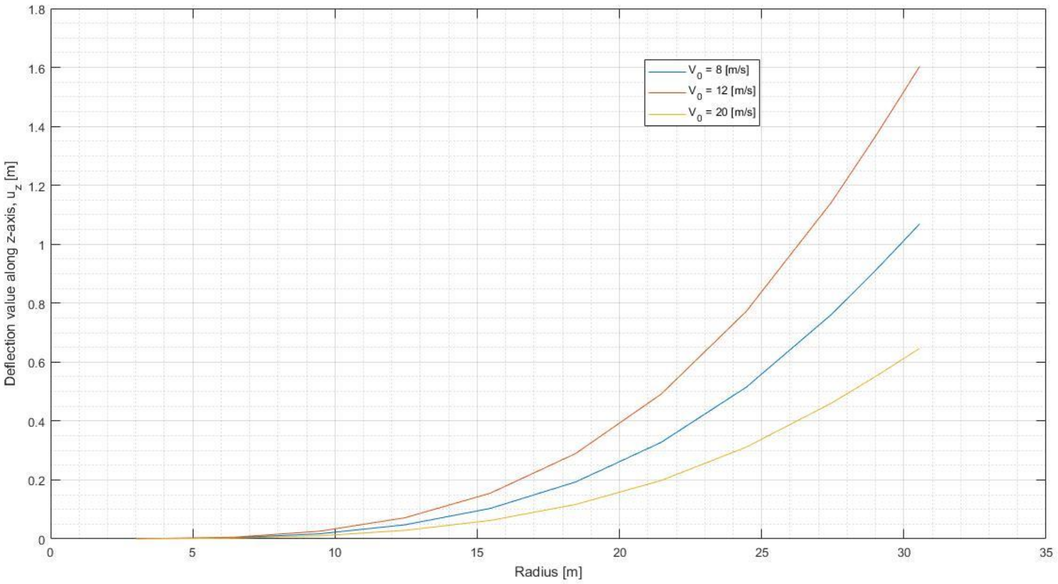


Fig.9 - deflection values and angles along the beam cantilever (u_z)

From the two plots, it's possible to see that the value of deflection increases along the blade: at the clamped/fixed side (R=3[m]) the deflections are equal to zero, at the free tip (R=30.55[m]) the deflections are at their highest.

The deflection values, along u_y, are the highest for wind speed of V_0=12[m/s], with θ_p=0°, which goes from zero (given in the boundary conditions) to 0.0514[m] at R=30.55[m].
While at V_0=20[m/s] the deflection is the lowest, it goes from zero (at R=3[m]) to 0.0131[m] (at R=30.55 [m]).

Lower values of deflection at V_0=20 m/s are due to the variation of the pitch angle. This shows the criticality of pitching of the blade, which lowers the external loads acting on the blade. This is a direct consequence of the angle of attack decreasing due to pitching, which decreases the lift and drag, and hence the loads acting on the blade, which decreases the deflections.

The deflections along u_z, as u_y, are the highest for V_0=12[m/s] and the lowest for V_0=20[m/s], and the graphs have also the same shape.
The tip deflection value at V_0=12[m/s] is 1.6030[m]; while the tip deflection at V_0=20[m/s] is 0.6466[m].

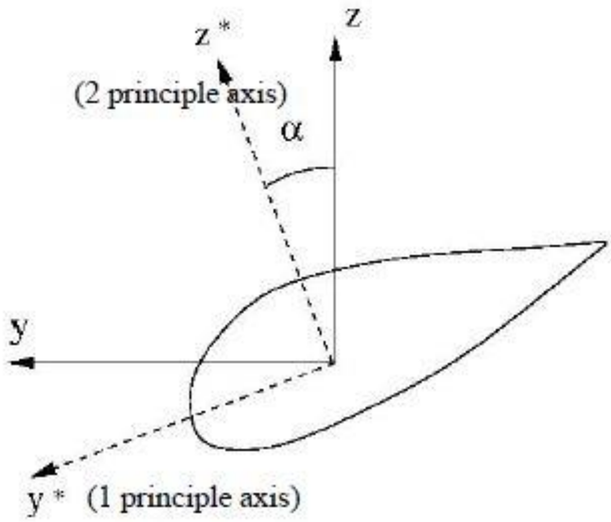
It is possible to notice that the deflections along z axe are higher, compared to the ones along u_y due to the fact that external loads p_z applied are higher than p_y.

To conclude, with increasing wind speeds, if the pitch angle of the blade is constant, the external loads acting on the blade increase, and as a consequence the deflections increase. On the other hand, it is possible to decrease the deflections by varying the pitch angle of the blade.

3. Q#3

For this question we are going to analyze the natural frequencies of the blade and its modal shapes.

We'll compute the frequencies for deflections in both flap-wise (2° principle axis) and edge-wise (1° principle axis) directions.



The algorithm used in MATLAB to compute such frequencies and modal shapes follows these steps:

- Compute the deflections for an unitary load for each blade section and for each direction p_y and p_z. The loads and deflections must be in the form

$$\mathbf{u} = \begin{bmatrix} u_{2y} \\ u_{2z} \\ \vdots \\ u_{iy} \\ u_{iz} \\ \vdots \\ u_{Ny} \\ u_{Nz} \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} p_{2y} \\ p_{2z} \\ \vdots \\ p_{iy} \\ p_{iz} \\ \vdots \\ p_{Ny} \\ p_{Nz} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_i \\ \vdots \\ \vdots \\ \vdots \\ p_M \end{bmatrix} \qquad \mathbf{F} \cdot \mathbf{p} = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1M} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ a_{M1} & \cdots & a_{Mi} & \cdots & a_{MM} \end{bmatrix} \cdot \mathbf{p} = \mathbf{u}$$

- Compute the matrix F (flexibility matrix) following the equation above. (The i^{th} column of F is the deflection vector obtained with $p_i = 1$ and $p_k = 0 \; where \; k \neq i$)

- Compute the mass matrix M, which is a diagonal matrix such that

$$\mathbf{p} = \omega^2 \mathbf{M} \cdot \mathbf{u} \quad \mathbf{M} = \begin{bmatrix} m_2 & & & & & & & 0 \\ & m_2 & & & & & & \\ & & \ddots & & & & & \\ & & & m_i & & & & \\ & & & & m_i & & & \\ & & & & & \ddots & & \\ & & & & & & m_N & \\ 0 & & & & & & & m_N \end{bmatrix}$$

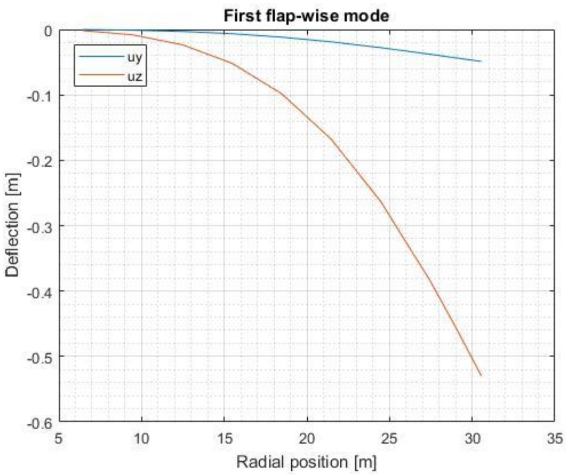
- Find the eigenvalues and eigenvectors of the matrix $F * M$. We used the built in function "eigs". $matlab\ command$: $[V, D] = eigs(F * M, 6)$

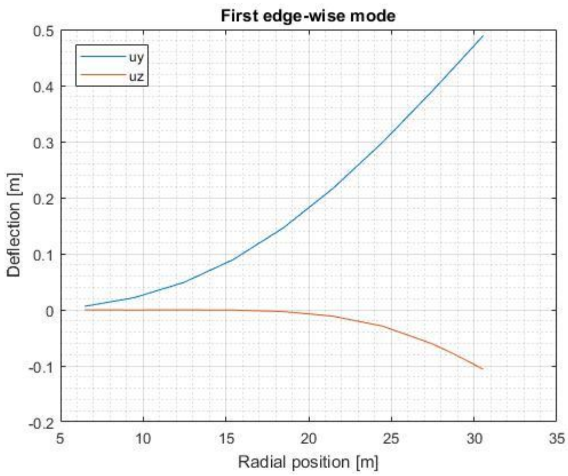
- The eigenvalues corresponds to $\frac{1}{\omega^2}$.
- The eigenvectors are the modal shapes.

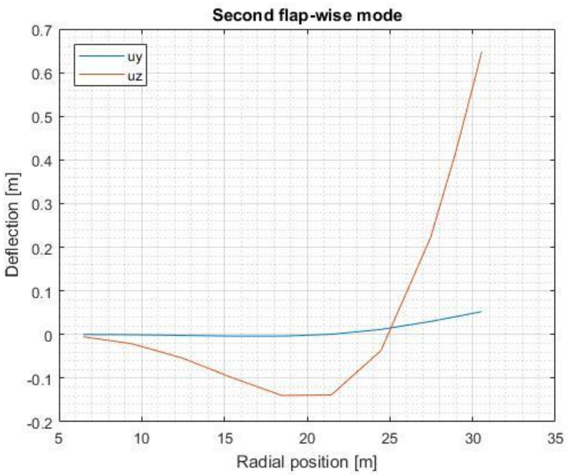
The resulting eigenfrequencies are summarized in the following table

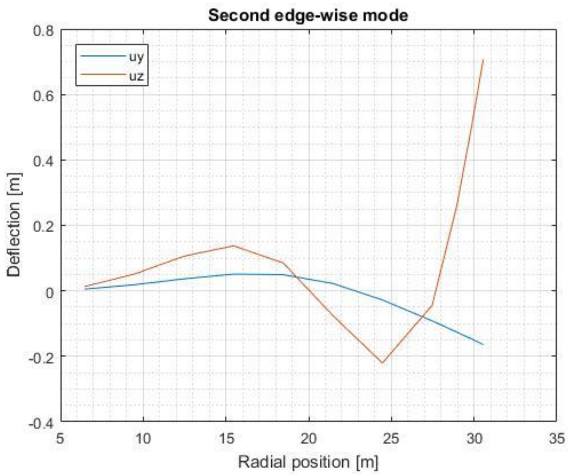
Eigenmode	**Eigenfrequency [Hz]**
First flap-wise	1.1960
First edge-wise	2.4076
Second flap-wise	3.4575
Second edge-wise	7.5760
Third flap-wise	8.3633
Third edge-wise	13.8048

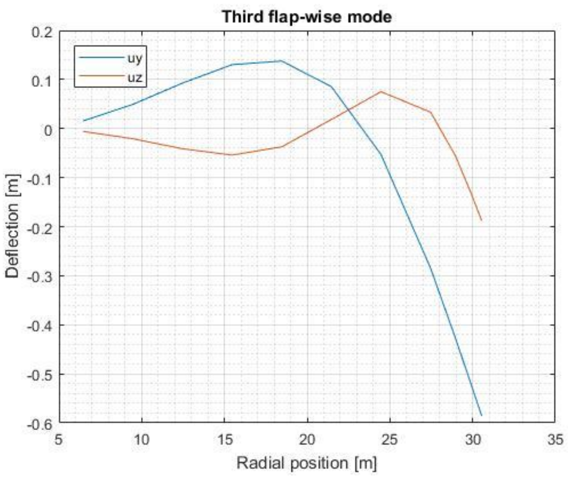
The modal shapes are presented in the following figures.

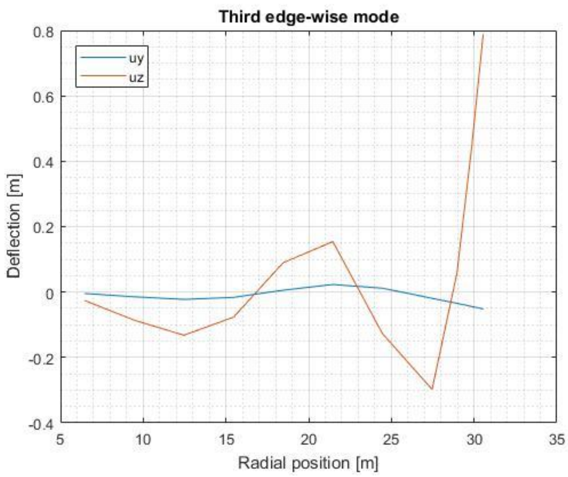












As we can see from the table, we have eigenfrequencies in a range from ~1Hz to ~14Hz. These frequencies play an important role in blade design due to the fact that an external force exciting the blade over these frequencies will cause a substantial amount of fatigue.

Looking at the plots, it is clear that higher modal shapes may bring the structure to bigger deflections. Higher modal shapes also have higher eigenfrequency and normally Kaimal spectrums don’t include frequencies that high. The fatigue due to these modal shapes should than be minimal.

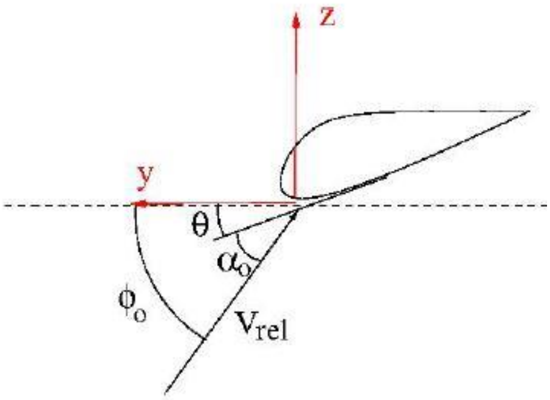
4. Q#4

Since eigenmodes oscillate accordingly to their eigenfrequencies, it would be of interest to know if the wind would actually allow the blade to oscillate at such frequencies.

For this question we are going to compute the work done by the wind when the blade oscillate following the equation of motion: $x(t) = \sin(2 * t)$. If the work is negative, the blade will tend to oscillate as described (If positive, it will not).

The blade is assembled on the rotor so that its chord line forms an angle (θ) of 1 degree with the rotor plane. The flow angle (Φ_0) in consideration is 8 degrees. The wind speed for this analysis is 70 m/s.

The blade will oscillate in both the y direction and z direction as described in the following figure.



This means that we'll have two values for the work done by the wind, one for each direction of motion.

The steps implemented in MATLAB are as follows:

$$\text{Forced vibration in z-direction} \quad x(t) = A\sin(\omega t)$$

$$V_{rel,y} = -V_{rel}\cos\phi_o$$

$$V_{rel,z} = V_{rel}\sin\phi_o - \dot{x}(t) = V_{rel}\sin\phi_o - \omega A\cos(\omega t)$$

$$\tan\phi(t) = \frac{V_{rel,z}}{-V_{rel,y}}$$

$$\alpha(t) = \phi(t) - \theta$$

$$C_l(\alpha(t)),\ C_d(\alpha(t))$$

$$C_z(t) = C_l(t)\cos(\phi(t)) + C_d(t)\sin(\phi(t))$$

$$p_z = \tfrac{1}{2}\rho V_{rel}^2 c C_z$$

$$W = \int_0^T p_z \dot{x}(t)\,dt$$

The algorithm used to compute the work for the motion in the y-direction are the same except that now the relative velocities are computed as:

$$V_{rel,y} = -V_{rel} * \cos(\Phi_0) + \omega * A * \cos(\omega * t)$$
$$V_{rel,z} = V_{rel} * \sin(\Phi_0)$$

The next table shows the work done by the wind in one cycle.

	z-direction	y-direction
Work [J]	-2076.8	290.83

As we can see the work is negative for the motion in the z-direction, meaning that the conditions described at the beginning of this question are so that the blade will oscillate following the equation $x(t) = \sin(2 * t)$.

The procedure explained for this question is significant to analyze the likelihood of a blade to oscillate when a wind of a given speed is blowing, in order to avoid critical design failures.

Conclusion

The report aimed the analysis of the structure of a blade of a wind turbine.

We have studied the Bernoulli Beam Theory for bending beam for different values of incoming wind speed in order to do a calculation of blade deflection.

From the obtained results shown in the plots and from the numerical values reported in tables, we figured out how much the relative velocity of the wind and the natural frequencies affects the total fatigue of the structure of a rotor in a wind turbine.

Fig.10 – Example of damage due to fatigue on a blade of a wind turbine

In particular, we can say that the analysis done in the report help us in deciding on what kind of a rotor blade is to be used in different ranges of wind speeds, and the pitch regulation which is done continuously to keep the power limited.

This is essential in making sure that the deflection is not too high, so as to avoid contact with the tower, fatigue or failure due to instability.

Appendix

1. MATLAB Codes

```
% 46300 - ASSIGNMENT 3 - Structure
%
%_________________________________________________________________________
clear all;
close all;
clc;
%_________________________________________________________________________
% VARIABLES PART

% General Datas
r_v=[3 6.46 9.46 12.46 15.46 18.46 21.46 24.46 27.46 28.96 29.86 30.55];    % r_v = Vector of
Distances on blade [m]
% EI1_v, EI2_v = distribution vector of stiffness on the first and second axis of the blade
EI1_v=[1.7e9 5.6657e8 2.3568e8 1.1916e8 5.9832e7 2.9763e7 1.3795e7 5.3929e6 1.5133e6 6.3600e5
3.5733e5 1.5917e5];
EI2_v=[1.7e9 1.4003e9 8.5164e8 5.2511e8 3.2937e8 2.0720e8 1.2099e8 5.9935e7 2.4543e7 1.4047e7
1.0060e7 7.2248e6];
mass_v=[3.3e2 3.3719e2 2.7564e2 2.2902e2 1.9140e2 1.6692e2 1.5947e2 8.4519e1 4.7877e1 3.3029e1
2.5357e1 1.9990e1];
Beta_v=[0.0 9.11 7.90 6.71 5.71 4.74 3.65 2.40 0.90 0.06 -0.44 -0.52];      % Beta_v = Vector of
blade twist values at blade sections [degree]
%
N=length(r_v);                                                             % N = number of
elements [-]
%
rho = 1.225;                                                               % rho = Air density
[kg/(m^3)]
%_________________________________________________________________________
% Q#1
%
% VARIABLES PART
EI1_cost=linspace(3e7,3e7,N);       % constant stiffness [Nm^2]
EI2_cost=linspace(3e7,3e7,N);
theta_p_in=0;                       % theta_p = pitch angle [rad]
%we consider that the principle axes are located along the chordline

beta=zeros(N);
p_z=linspace(2000,2000,N);
p_y=linspace(2000,2000,N);

R=27.55;                            % R = r_v(N) - r_v(1) = 30.55 - 3 = 27.55[m]
x=r_v-3;                            % x = scaling of r_v on blade radial-axis [m]


% COMPUTATION PART
% Steady deflections with zero accelerations

% Boundary conditions cantilever beam
T_y(N)=0;
T_z(N)=0;
M_y(N)=0;
M_z(N)=0;
for i=N:-1:2
```

```
    % share forces
    T_y(i-1)=T_y(i)+(1/2)*(p_y(i-1)+p_y(i))*(x(i)-x(i-1));
    T_z(i-1)=T_z(i)+(1/2)*(p_z(i-1)+p_z(i))*(x(i)-x(i-1));
    % bending moments
    M_y(i-1)=M_y(i)-T_z(i)*(x(i)-x(i-1))-((1/6)*p_z(i-1)+(1/3)*p_z(i))*(x(i)-x(i-1))^2;
    M_z(i-1)=M_z(i)+T_y(i)*(x(i)-x(i-1))+((1/6)*p_y(i-1)+(1/3)*p_y(i))*(x(i)-x(i-1))^2;
end

% The simple relationship between curvature and bending moments only valid about principle axes
for i=1:N
    % bending moments of first (M_1) and second (M_2) principal axis
    M_1(i)=M_y(i)*cosd(abs(beta(i)+theta_p_in))-M_z(i)*sind(abs(beta(i)+theta_p_in));
    M_2(i)=M_y(i)*sind(abs(beta(i)+theta_p_in))+M_z(i)*cosd(abs(beta(i)+theta_p_in));
    % relationship between the deflections and internal loads
    k_1(i)=M_1(i)/EI1_cost(i);
    k_2(i)=M_2(i)/EI2_cost(i);
    % Curvatures transfered back to blade coordinate system
    k_z(i)=-k_1(i)*sind(abs(beta(i)+theta_p_in))+k_2(i)*cosd(abs(beta(i)+theta_p_in));
    k_y(i)=k_1(i)*cosd(abs(beta(i)+theta_p_in))+k_2(i)*sind(abs(beta(i)+theta_p_in));
end

% The blade coordinate system follows the tip and pitch is the angle between tip chord and rotor
plane.
% But we want to take pitch into account. Thus, we integrated it from the root to the tip.
% Boundary conditions cantilever beam (calculated at the root of the blade)
theta_y(1)=0;
theta_z(1)=0;
u_y(1)=0;
u_z(1)=0;
for i=1:(N-1)
    % deflection angles
    theta_y(i+1)=theta_y(i)+1/2*(k_y(i+1)+k_y(i))*(x(i+1)-x(i));
    theta_z(i+1)=theta_z(i)+1/2*(k_z(i+1)+k_z(i))*(x(i+1)-x(i));
    % values of deflection (u_y and u_z) along y and z axes
    u_y(i+1)=u_y(i)+theta_z(i)*(x(i+1)-x(i))+((1/6)*k_z(i+1)+1/3*k_z(i))*(x(i+1)-x(i))^2;
    u_z(i+1)=u_z(i)-theta_y(i)*(x(i+1)-x(i))-((1/6)*k_y(i+1)+1/3*k_y(i))*(x(i+1)-x(i))^2;
end
% The values of u_y and u_z are equal beacuse we chose p_y=p_z=2000.

u_x(1)=0;
for v=1:N
    psi(v)=(R-x(v))/R;                                    % psi = strain on the blade [-]
    u_x(v)=((p_y*R^4)/(24*EI1_cost))*(psi(v)^4-4*psi(v)+3);
end


% PLOTS PART
figure;          % plot of deflection along the beam cantilever
plot (r_v,u_y);
xlabel ('Radius [m]');
% xlim ([3,30.55]);
ylabel ('Deflection value u_y= u_z [m]');
grid on;
grid minor;
hold on;
plot (r_v,u_x);
legend('u_y = u_z','u_x = u analytical');
%__________________________________________________________________________
% Q#2
```

```
%
% VARIABLES PART
V_0=[8 12 20];
theta_p=[0;0;15.37];
p_tan=[170;490;200];
p_norm=[2000;3000;1200];

theta_p=repmat(theta_p,[1 N]);


% COMPUTATION PART
for v=1:length(V_0)
    for i = 2:1:N
        p_y(v,i) = p_tan(v).*cosd(theta_p(v))-p_norm(v).*sind(theta_p(v));
        p_z(v,i) = p_tan(v).*sind(theta_p(v))+p_norm(v).*cosd(theta_p(v));
    end
    for i=N:-1:2
        T_y(v,N)=0;
        T_z(v,N)=0;
        M_y(v,N)=0;
        M_z(v,N)=0;
        T_y(v,i-1)=T_y(v,i)+(1/2)*(p_y(v,i-1)+p_y(v,i))*(x(i)-x(i-1));
        T_z(v,i-1)=T_z(v,i)+(1/2)*(p_z(v,i-1)+p_z(v,i))*(x(i)-x(i-1));
        M_y(v,i-1)=M_y(v,i)-T_z(v,i)*(x(i)-x(i-1))-((1/6)*p_z(v,i-1)+(1/3)*p_z(v,i))*(x(i)-x(i-
1))^2;
        M_z(v,i-1)=M_z(v,i)+T_y(v,i)*(x(i)-x(i-1))+((1/6)*p_y(v,i-1)+(1/3)*p_y(v,i))*(x(i)-x(i-
1))^2;
    end
    for i=1:N
        M_1(v,i)=M_y(v,i)*cosd((beta(i)+theta_p_in))-M_z(v,i)*sind((beta(i)+theta_p_in));
        M_2(v,i)=M_y(v,i)*sind((beta(i)+theta_p_in))+M_z(v,i)*cosd((beta(i)+theta_p_in));
        k_1(v,i)=M_1(v,i)/EI1_v(i);
        k_2(v,i)=M_2(v,i)/EI2_v(i);
        k_z(v,i)=-k_1(v,i)*sind((beta(i)+theta_p_in))+k_2(v,i)*cosd((beta(i)+theta_p_in));
        k_y(v,i)=k_1(v,i)*cosd((beta(i)+theta_p_in))+k_2(v,i)*sind((beta(i)+theta_p_in));
    end
    for i=1:(N-1)
        theta_y(v,1)=0;
        theta_z(v,1)=0;
        u_y(v,1)=0;
        u_z(v,1)=0;
        theta_y(v,i+1)=theta_y(v,i)+1/2*(k_y(v,i+1)+k_y(v,i))*(x(i+1)-x(i));
        theta_z(v,i+1)=theta_z(v,i)+1/2*(k_z(v,i+1)+k_z(v,i))*(x(i+1)-x(i));
        u_y(v,i+1)=u_y(v,i)+theta_z(v,i)*(x(i+1)-x(i))+((1/6)*k_z(v,i+1)+1/3*k_z(v,i))*(x(i+1)-
x(i))^2;
        u_z(v,i+1)=u_z(v,i)-theta_y(v,i)*(x(i+1)-x(i))-((1/6)*k_y(v,i+1)+1/3*k_y(v,i))*(x(i+1)-
x(i))^2;
    end
end


% PLOTS PART
figure;         % deflection values and angles along the beam cantilever
plot (r_v,u_y);
xlabel ('Radius [m]');
ylabel ('Deflection value along y-axis, u_y [m]');
grid on;
grid minor;
legend('V_0 = 8 [m/s]','V_0 = 12 [m/s]','V_0 = 20 [m/s]');
```

```
figure;
plot (r_v,u_z);
xlabel ('Radius [m]');
ylabel ('Deflection value along z-axis, u_z [m]');
grid on;
grid minor;
legend('V_0 = 8 [m/s]','V_0 = 12 [m/s]','V_0 = 20 [m/s]');

figure;
plot (r_v,theta_y);
xlabel ('Radius [m]');
ylabel ('Deflection angles along y-axis, theta_y [rad]');
grid on;
grid minor;
legend('V_0 = 8 [m/s]','V_0 = 12 [m/s]','V_0 = 20 [m/s]');

figure;
plot (r_v,theta_z);
xlabel ('Radius [m]');
ylabel ('Deflection angles along z-axis, theta_z [rad]');
grid on;
grid minor;
legend('V_0 = 8 [m/s]','V_0 = 12 [m/s]','V_0 = 20 [m/s]');
%_________________________________________________________________________
% Q#3
%
% VARIABLES PART
EI1=[1.7e9 5.6657e8 2.3568e8 1.1916e8 5.9832e7 2.9763e7 1.3795e7 5.3929e6 1.5133e6 6.3600e5
3.5733e5 1.5917e5];
EI2=[1.7e9 1.4003e9 8.5164e8 5.2511e8 3.2937e8 2.0720e8 1.2099e8 5.9935e7 2.4543e7 1.4047e7
1.0060e7 7.2248e6];
R=[3 6.46 9.46 12.46 15.46 18.46 21.46 24.46 27.46 28.96 29.86 30.55];
beta=[0 9.11 7.90 6.71 5.71 4.74 3.65 2.40 0.90 0.06 -0.44 -0.52];
m=[3.3*10^2 3.3719*10^2 2.7564*10^2 2.2902*10^2 1.9140*10^2 1.6692*10^2 1.5947*10^2 84.519 47.877
33.029 25.357 19.990];

% COMPUTATION PART
N=length(R);
py_new=ones(1,N);
pz_new=ones(1,N);
pitch=zeros(1,N);
F=zeros(2*N-2,2*N-2);
tempuy=0;
tempuz=0;

for j=1:2*N-2
    p=zeros(1,2*N-2);
    u=zeros(1,2*N-2);
    p(j)=1;
    py=0;
    pz=0;
    for l=1:N-1
        py=[py p(1+(l-1)*2)];
        pz=[pz p(2*l)];
    end
    Ty=zeros(1,N);
    Tz=zeros(1,N);
    My=zeros(1,N);
```

```
    Mz=zeros(1,N);
    M1=zeros(1,N);
    M2=zeros(1,N);
    k1=zeros(1,N);
    k2=zeros(1,N);
    ky=zeros(1,N);
    kz=zeros(1,N);
    for i=N-1:-1:1
        Ty(i)=Ty(i+1)+0.5*(py(i)+py(i+1))*(R(i+1)-R(i));
        Tz(i)=Tz(i+1)+0.5*(pz(i)+pz(i+1))*(R(i+1)-R(i));
        My(i)=My(i+1)-Tz(i+1)*(R(i+1)-R(i))-(1/6*pz(i)+1/3*pz(i+1))*(R(i+1)-R(i))^2;
        Mz(i)=Mz(i+1)+Ty(i+1)*(R(i+1)-R(i))+(1/6*py(i)+1/3*py(i+1))*(R(i+1)-R(i))^2;
        M1(i)=My(i)*cosd(beta(i)+pitch(i))-Mz(i)*sind(beta(i)+pitch(i));
        M2(i)=My(i)*sind(beta(i)+pitch(i))+Mz(i)*cosd(beta(i)+pitch(i));
        k1(i)=M1(i)/EI1(i);
        k2(i)=M2(i)/EI2(i);
        kz(i)=-k1(i)*sind(beta(i)+pitch(i))+k2(i)*cosd(beta(i)+pitch(i));
        ky(i)=k1(i)*cosd(beta(i)+pitch(i))+k2(i)*sind(beta(i)+pitch(i));

    end
    tetay=zeros(1,N);
    tetaz=zeros(1,N);
    uz=zeros(1,N);
    uy=zeros(1,N);
    for i=1:N-1
        tetay(i+1)=tetay(i)+0.5*(ky(i+1)+ky(i))*(R(i+1)-R(i));
        tetaz(i+1)=tetaz(i)+0.5*(kz(i+1)+kz(i))*(R(i+1)-R(i));
        uy(i+1)=uy(i)+tetaz(i)*(R(i+1)-R(i))+(1/6*kz(i+1)+1/3*kz(i))*(R(i+1)-R(i))^2;
        uz(i+1)=uz(i)-tetay(i)*(R(i+1)-R(i))-(1/6*ky(i+1)+1/3*ky(i))*(R(i+1)-R(i))^2;
    end
    for l=1:N-1
        u(1+(l-1)*2)=uy(1+l);
        u(2*l)=uz(1+l);
    end
    F(:,j)=u;
end

M=zeros(2*N-2,2*N-2);
for l=1:N-1
    M(1+(l-1)*2,1+(l-1)*2)=m(1+l);
    M(2*l,2*l)=m(1+l);
end

[V,D]=eigs(F*M,6);
for i=1:6
    omega(i)=sqrt(1/D(i,i)); %[rad/s]
    ef(i)=omega(i)/(2*pi); %[Hz]
end
%omega as:
%[1° mode flap-wise,1° mode edge-wise,2° mode flap-wise,2° mode edge-wise, ...]
%ef like omega

uy=zeros(6,N-1);
uz=zeros(6,N-1);
for i=1:6
    for l=1:N-1
        uy(i,l)=V(1+(l-1)*2,i);
        uz(i,l)=V(l*2,i);
    end
```

```matlab
end

% PLOTS PART
figure
plot(R(2:end),uy(1,:),R(2:end),uz(1,:))
title('First flap-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')

figure
plot(R(2:end),uy(2,:),R(2:end),uz(2,:))
title('First edge-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')

figure
plot(R(2:end),uy(3,:),R(2:end),uz(3,:))
title('Second flap-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')

figure
plot(R(2:end),uy(4,:),R(2:end),uz(4,:))
title('Second edge-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')

figure
plot(R(2:end),uy(5,:),R(2:end),uz(5,:))
title('Third flap-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')

figure
plot(R(2:end),uy(6,:),R(2:end),uz(6,:))
title('Third edge-wise mode')
grid on
grid minor
xlabel('Radial position [m]')
ylabel('Deflection [m]')
legend('uy','uz','Location','northwest')
%_________________________________________________________________________
% Q#4
%
```

```
c=1.5;                  % [m]
phi0=deg2rad(8);        % [rad]
teta=deg2rad(1);        % [rad]
Vrel=70;                % [m/s]
rho=1.225;              % [Kg/m^3]
taj=load('tjaere11.dat');
alphataj=deg2rad(taj(:,1));
Cltaj=taj(:,2);
Cdtaj=taj(:,3);
clear taj
omega=2;
% motion along axis-z
t=linspace(0,pi,1000);
Vrely=-Vrel*cos(phi0);
Vrelz=Vrel*sin(phi0)-omega*cos(omega*t);
phi=atan(Vrelz/(-Vrely));
alpha=phi-teta;
Cl=interp1(alphataj,Cltaj,alpha,'linear');
Cd=interp1(alphataj,Cdtaj,alpha,'linear');
Cz=Cl.*cos(phi)+Cd.*sin(phi);
pz=0.5*rho*Vrel^2*c*Cz;
Wz=trapz(t,pz*omega.*cos(omega*t));

%% motion along axis-y
Vrely=-Vrel*cos(phi0)+omega*cos(omega*t);
Vrelz=Vrel*sin(phi0);
phi=atan(Vrelz./(-Vrely));
alpha=phi-teta;
Cl=interp1(alphataj,Cltaj,alpha,'linear');
Cd=interp1(alphataj,Cdtaj,alpha,'linear');
Cy=Cl.*cos(phi)+Cd.*sin(phi);
py=0.5*rho*Vrel^2*c*Cy;
Wy=trapz(t,py*omega.*cos(omega*t));
```