Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

WIND POWER GENERATION AND OPTIMIZATION CONSIDERING THE WAKE EFFECT

MEMÒRIA

AUTOR: DAVID BIGORDÀ PIELLA

DIRECTOR: MARC CHEAH MAÑÉ

CONVOCATÒRIA: JUNY 2019

ETSEIB

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

UPC

ABSTRACT

This project, firstly, postulates the hypothesis that operating the different wind turbines of a wind power plant at a concrete performance and different from the optimal one, has as a result an increase on the wind speeds in the wakes and therefore, on the wind speed that reaches the downwind turbines shadowed by the forward ones, and consequently, a significant increase on the total electric generation of the farm. Previously, the aerodynamic theory that will be necessary to develop the project is reviewed, as well as the models that define the wind speeds depending on the wake effect. “Le Parc Éolien en mer de Fécamp” is also described as the real application case of study. The hypothesis of the optimization methodology is verified through two more simple cases and, finally, the real application case of study is considered with all his wind turbines at the optimal point, in order to calculate the total energy produced by the wind power plant including the wake effect and thus, demonstrating the global loss of energy production by means of this conventional operation strategy.

RESUMEN

Este proyecto primeramente plantea la hipótesis de operar los diferentes aerogeneradores de un parque eólico a un rendimiento concreto y distinto del óptimo tiene como resultado un aumento de las velocidades del viento en las estelas y por lo tanto en la velocidad del viento que llega a las turbinas que quedan eclipsadas por otras turbinas y, por consiguiente, un aumento significativo de la generación eléctrica total del parque. Previamente se repasa la teoría aerodinámica de los aerogeneradores, los modelos que definen las velocidades del viento según las estelas que producen las turbinas y se describe el caso real de estudio de este proyecto: "Le Parc Éolien en mer de Fécamp". Se verifica la hipótesis de optimización mediante dos casos más sencillos y por último se considera el caso real de estudio con todos los aerogeneradores en su particular punto óptimo, en el que se calcula la energía total del parque teniendo en cuenta el efecto de estela y demostrando así, la pérdida global de energía.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CFD: Computational Fluid Dynamics

ETS: The European Union Emissions Trading System (EU ETS)

EWEA: The European Wind Energy Association

GE Energy: General Electric Power

HVAC: Heating, Ventilation, and Air Conditioning

IEC: International Electrotechnical Commission

MATLAB: Matrix Laboratory

MPPT: Maximum Power Point Tracker

OWPP: Offshore Wind Power Plant

PMG: Permanent Magnet Generator

WPP: Wind Power Plant

WT: Wind Turbine

XLPE: Cross-linked polyethylene

1. Introduction and hypothesis

The aim of this project is, from a steady state point of view, to analyse the potential benefit of a Wind Power Plant (WPP) control strategy whose main objective is to maximise its total energy by taking into consideration the wake effect, depending on the operation of each wind turbine. Unlike the conventional approach in which each wind turbine operation is optimised individually to maximise its own energy capture, the proposed control strategy aims to optimise the whole system by operating some wind turbines at sub-optimum points. So that the wake effect within the WPP is reduced and, therefore, the total power generation is maximised. This global optimization will be analysed in two easier cases and once validated, it will be studied in a specific offshore Wind Power Plant, The *Parc Éolien en mer de Fécamp*, with 83 wind turbines in the north of France, which is thought to be run during the 2022.

To achieve this, a review of the aerodynamic theory of wind turbines is made. Then, an operation system is established in order to make use of this effect to improve the energy production of the whole park.

Nowadays, wind energy is an undeniable reality in our society, presents high levels of penetration in the energy mix of most major European countries and it is an unstoppable way in many of the economies developing the world. It is a clean and autochthonous energy that contributes to the productive economy and creates added value in those countries where it is present. At European level, where the energy policies of the countries are guided by community strategies, wind energy is postulated as a fundamental element to meet the objectives of reducing greenhouse gas emissions as well as the penetration of renewable energies and the reduction of energy dependence.

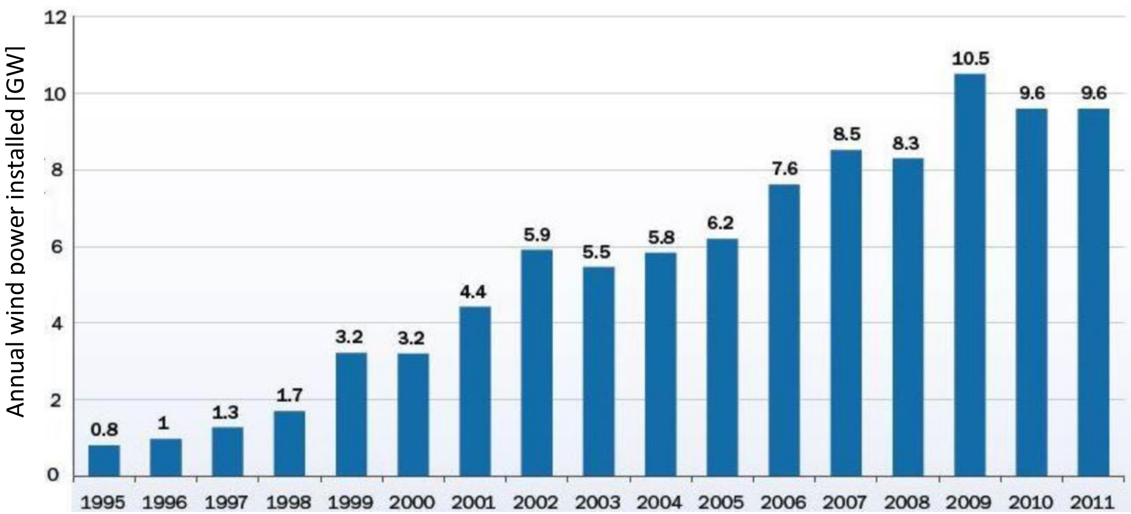


Figure 1.1: Evolution of the annual wind power installed in Europe. [EWEA, 2012], [1]

Due to the great growth it has experienced in recent years (see figure 1.1), wind energy is today the source of renewable energy with a greater power installed after the hydraulics. As it can be seen in the figure 1.2, it represents the 11% of all the electric power installed in the EU and currently generates around 6% of the electricity consumed.

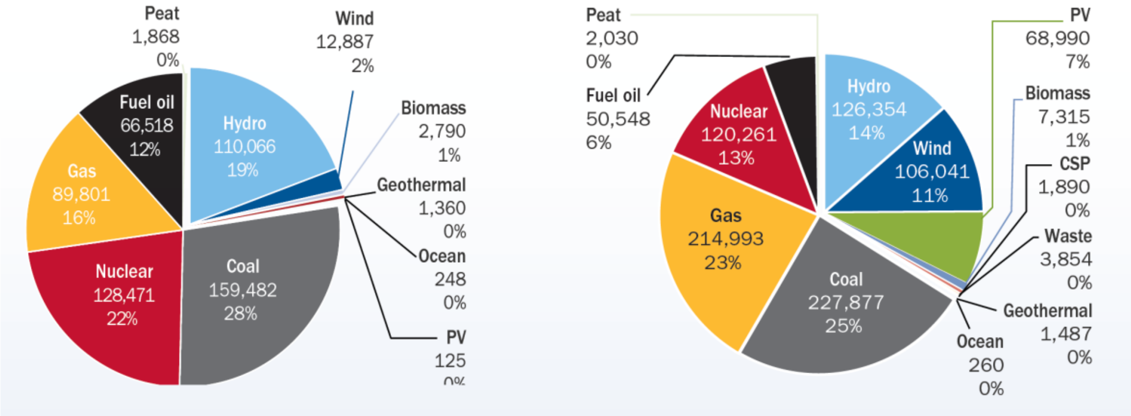


Figure 1.2: Installed power in the EU in 2000 and 2012. 2012, [ETS]

Although wind technology is already well established in its more traditional variants and locations, it is still an important field of research and development in aspects such as rotors, with a tendency to be increasingly larger; the more and more flexible and versatile generators and finally the locations. Due to the fact that certain areas are beginning to show some saturation, the repowering of parks with more modern and efficient machines as well as the installation of these in less conventional places, such as the case of the offshore wind, is becoming rather usual, as seen in figure 1.3.

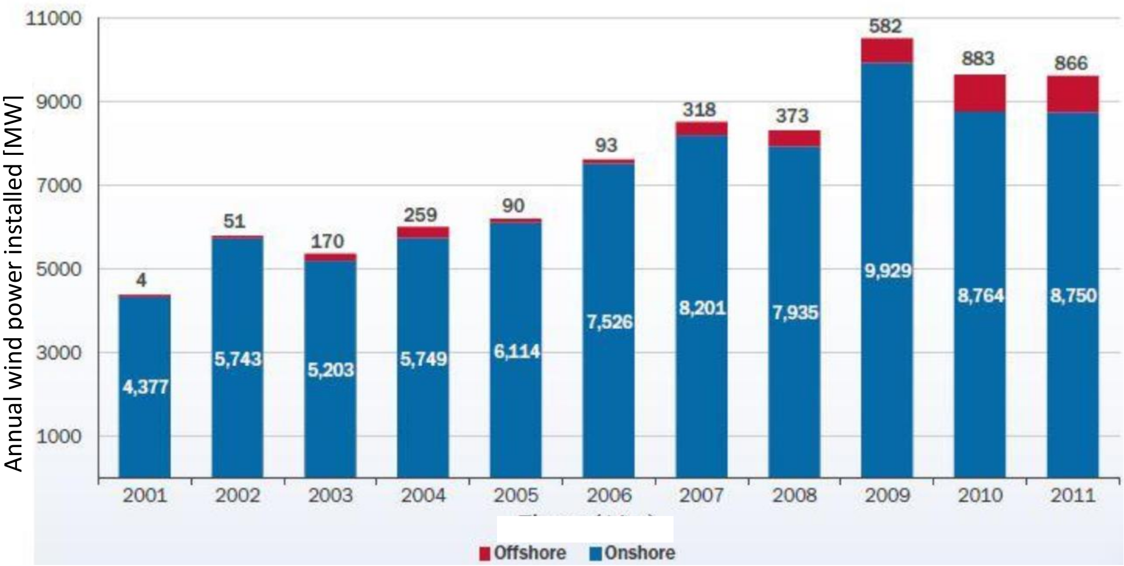


Figure 1.3: Growth of the annual offshore wind power installed in Europe. [1]

The main advantage of offshore wind is the large number of available sites that have better wind characteristics to those usually found in onshore sites. However, this technology has also a new series of problems such as the increase of installation costs, corrosion and transport of energy to the coast. Although today, there are enough technological solutions to overcome these disadvantages it is always essential to look for the most favourable locations. [6], [7], [8]

So definitely, there is an effect of shielding or wind shadowing between turbines when the wind blows in certain directions. This causes the power developed by the wind turbines of the back lines being lower than that of the forward ones. This effect together with the fact that every time there is a greater tendency to install generators with total power converters that offer a great control of the machine in operating it at variable speeds, has motivated the hypothesis that is exposed next.

The hypothesis is that it is possible to improve the energy production efficiency of a set of wind turbines if each of the turbines is operated at a different specific rotation speed taking into account the wake effects that these cause in the back wind turbines.

Hence, since an operation at a specific lower speed implies a greater availability of wind in the wake, it is said that operating the wind turbines of the forward lines at a specific speed lower than the one that would present a maximum power coefficient, will cause a greater availability of wind in the subsequent ones and, thus, the increase of the energy production in this ones will not only supply the deficit of the forward lines but will increase the total energy production.

Nowadays, the strategy to follow, with this variable speed machines, is to vary the angular speed of the rotor to keep the specific speed constant in the value that maximizes the power coefficient until the nominal power is reached, right when the turbine begins being operated at a fixed angular speed. This is done independently for each turbine because each one has an anemometer to measure the wind speed and an electronic system ruled by an MPPT (Maximum Power Point Tracker) system. The aim of this study is to develop an operation system that acts as MPPT at a park level and not only at a wind turbine level and, indeed, the increase of the energy production.

2. Description of the wind plant

2.1. Parc Éolien en mer de Fécamp

The *Parc Éolien en mer de Fécamp* is a French offshore wind farm located in the English Channel sea (49.892, 0.227), far enough to the coast (13 to 22 km of distance) and with an area of 67 km^2. It has **83** wind turbines, with a height of **175 meters** and **6 MW** of power capacity (498 MW of the total project capacity), and the model of the WT is called Alstom Haliade 150-6MW (GE Energy). With a 25 years' life expectation, it is thought to hold 352.711 homes powered annually and to reduce 712.829 tonnes of CO2 per year. [9]

This wind farm is thought to be turned on the 2022. The wind turbines are laid out according to defined alignments set by the professional fishermen to limit the impact on their activities and to reduce the landscape impact of the project.

Since 2007, an important work has been carried out with the local employment, the formation and the insertion in order to make benefit with the territory and its economic consequence. The construction of the gravitational foundations of wind turbines will enable the mobilization of 600 people on the port of Le Havre. Throughout the life of the park, about 25 years, the project will create hundreds of jobs in Fécamp to ensure maintenance operations. [2]

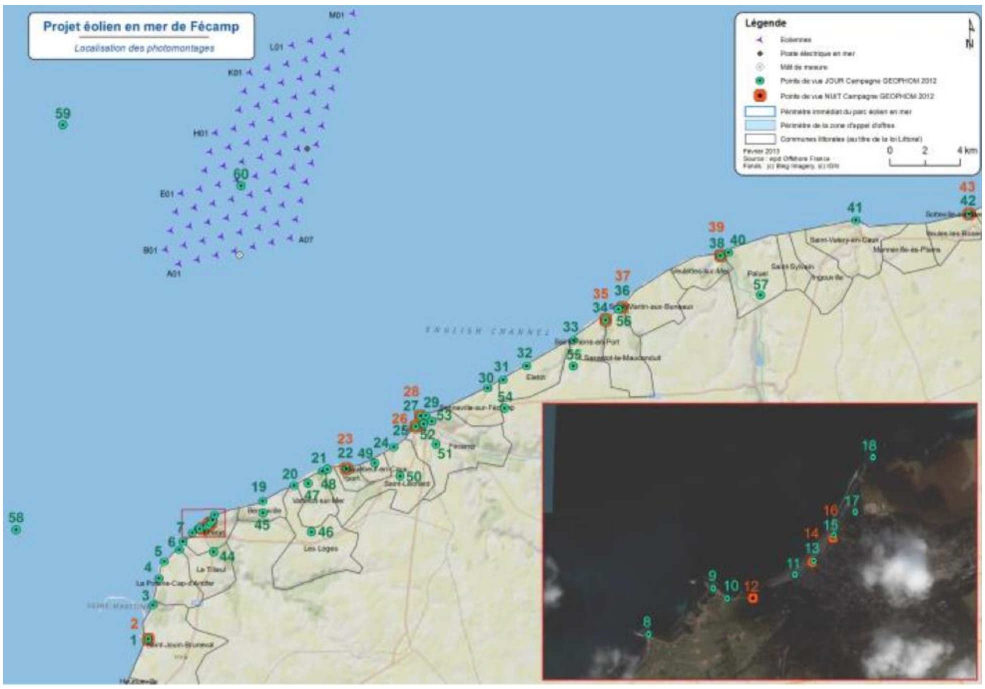


Figure 2.1: Map of the location of Fécamp. [2]

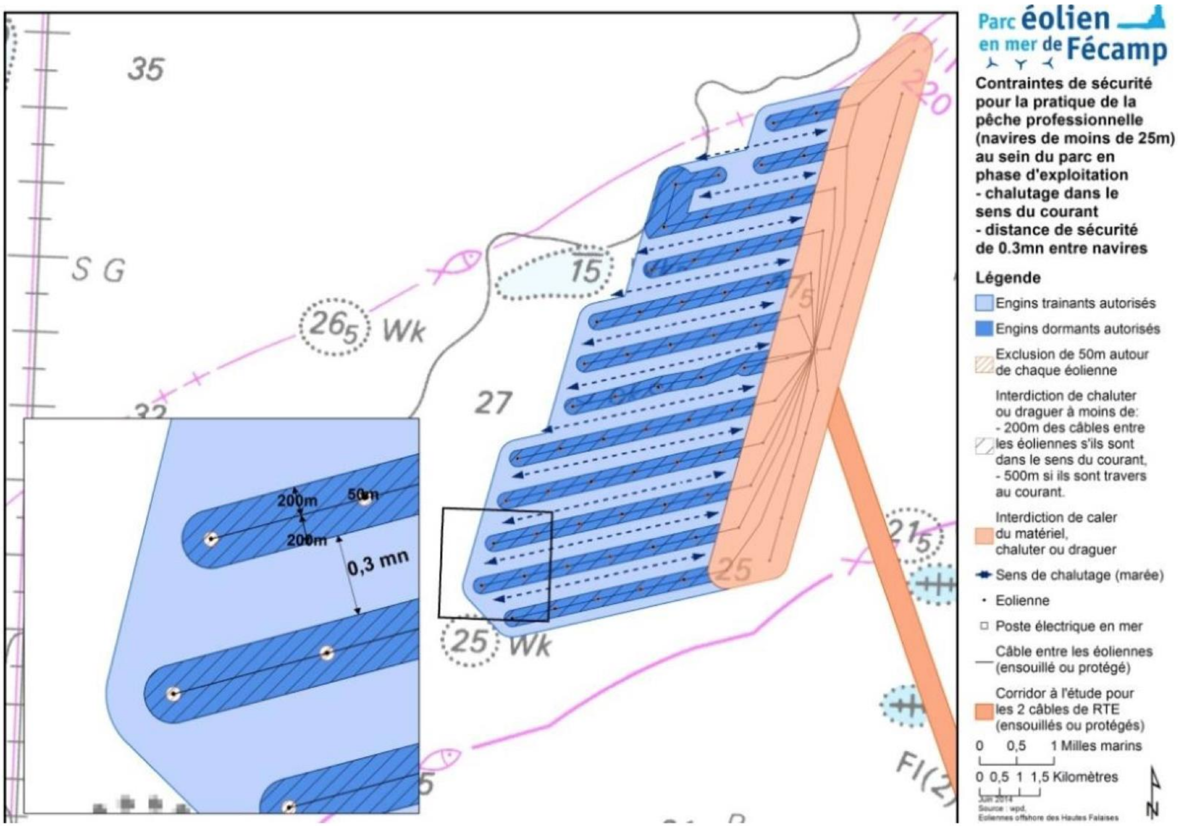


Figure 2.2: Layout of the Fécamp wind power plant. [2]

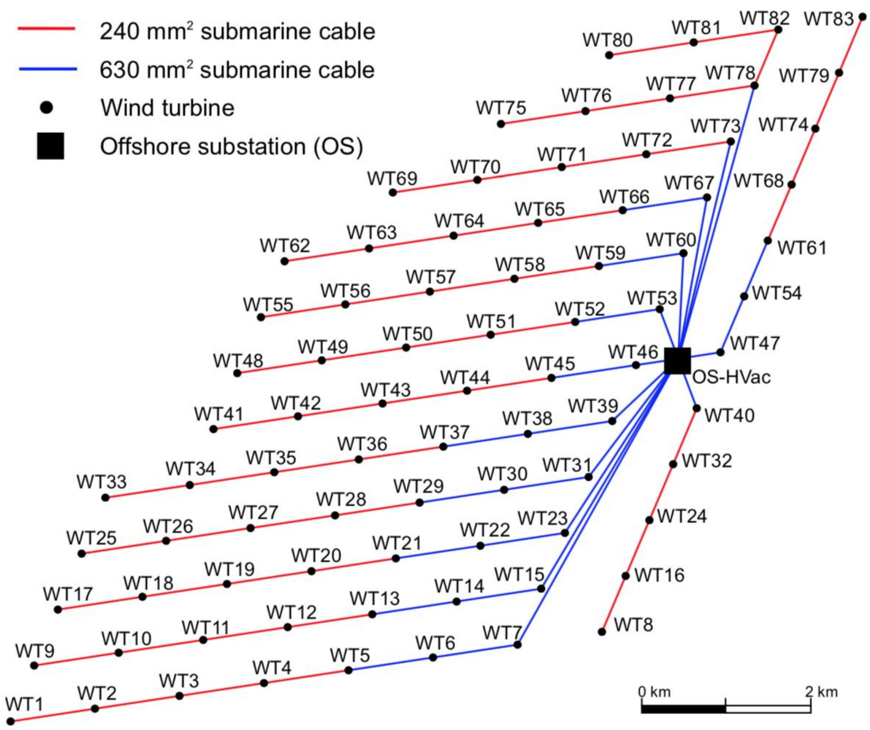


Figure 2.3: Layout of the Fécamp wind power plant. [3]

The turbines are arranged in 13 strings connected via submarine cables to the offshore HVac substation (OS-HVac). Two cross-sections are used for the cross-linked polyethylene (XLPE) submarine cables in the collection grid, 240 mm^2 and 630 mm^2, whereas the export cables are of 800 mm^2.

2.2. Wind data in the location

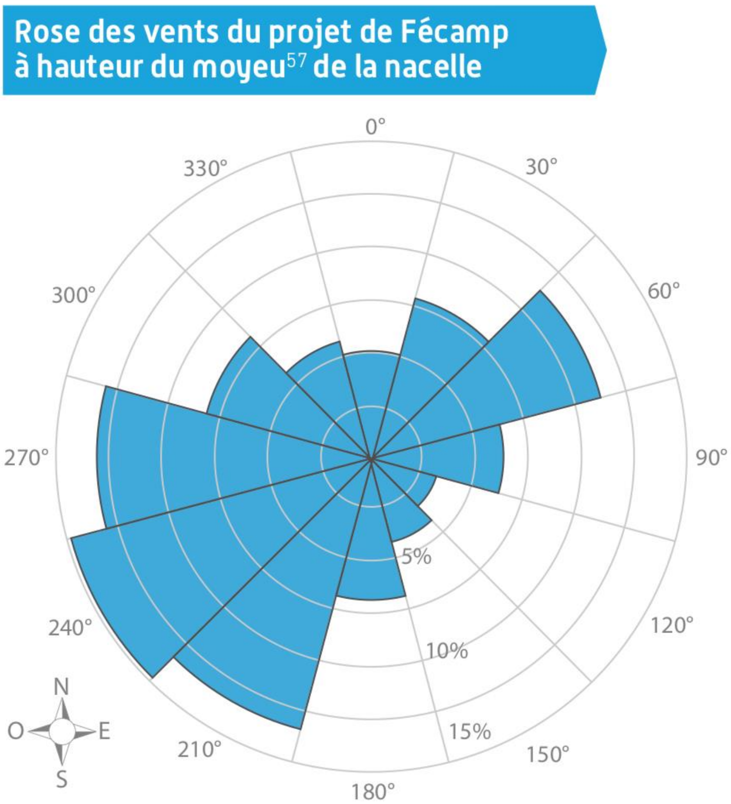


Figure 2.4: Fécamp wind rose at the hub height. [4]

At the sea of Fécamp project, the measured average wind speeds are greater than 9 meters per second (**32,5 km/h**).

The wind turbines are designed to operate at wind speeds between **3 m/s** (10 km/h) and **25 meters per second** (90 km/h). They operate at their full power from **12 meters per second** on (45 km/h). An anemometer and a weather vane on the *nacelle* enable them to continuously measure the wind speed and determine its direction. When the wind speed is sufficient, the rotor is then placed facing the wind in order to capture its energy in the best conditions. Beyond 90 km/h, the blades of the wind turbine turn around their axis to reduce wind resistance and stop working for safety reasons. So, as said, the cut-in wind speed is 3 m/s and the cut-out wind speed is 25 m/s. [4]

In the location of the wind plant, in Fécamp (49.892, 0.227), another wind rose is obtained from the Meteoblue web page, but this time measured at 10 meters of height. [5]

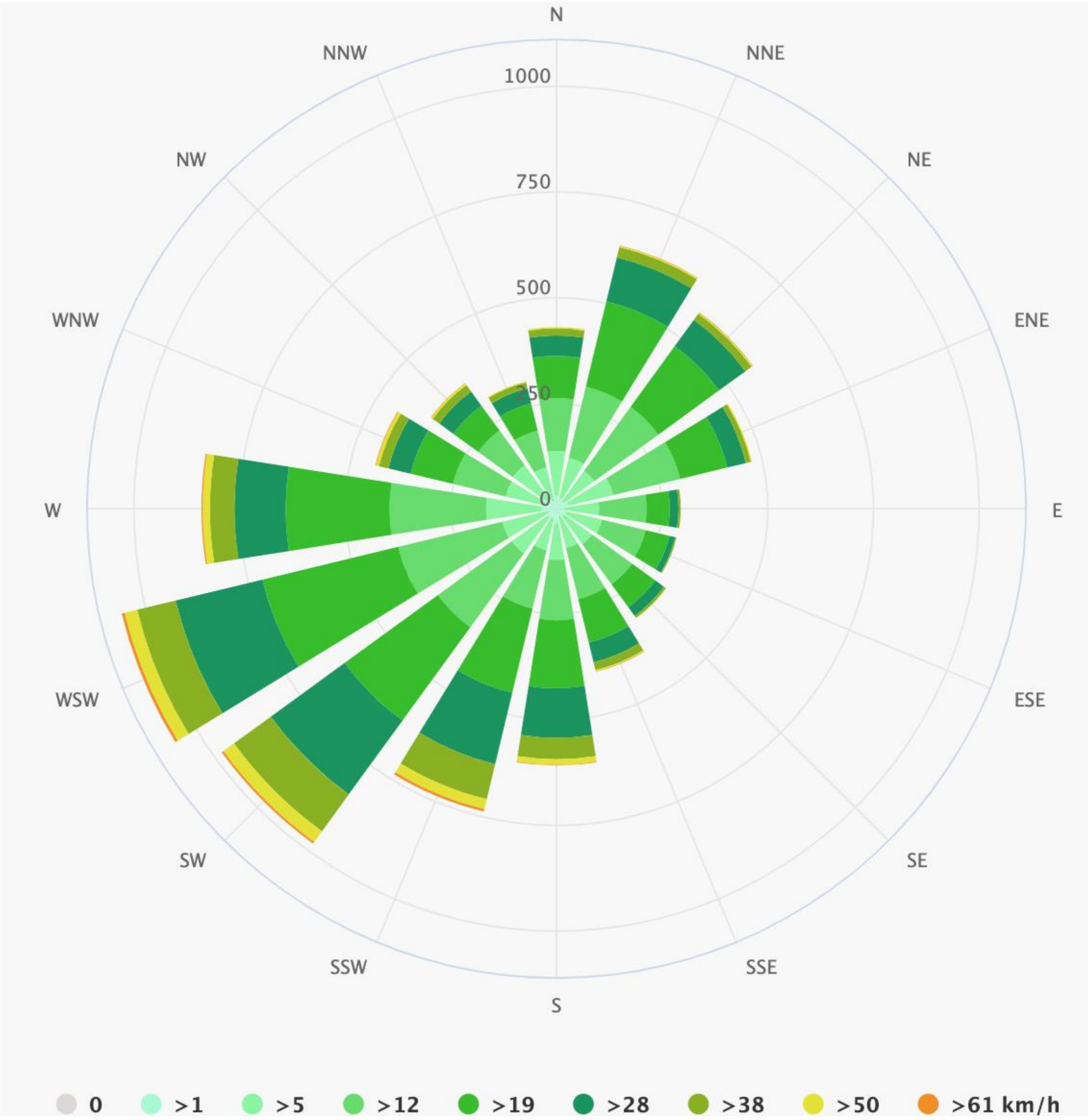


Figure 2.5: Fécamp wind rose at 10 meters' height. [5]

In this last figure, the data is based in 30 year hourly measurements and shows how many hours per year the wind blows from the indicated direction. The data in table format is shown here:

	Wind speed [m/s]								
H/YEAR	**0**	**>0,3**	**>1,6**	**>3,4**	**>5,5**	**>8,0**	**>10,8**	**>13,9**	**>17,2**
0	2	30	105	125	100	48	17	3	0
22,5	1	26	97	173	208	107	25	3	1
45	0	21	90	187	178	77	18	3	1
67,5	2	28	108	163	115	45	12	3	0

Wind direction [degrees]

90	0	18	83	112	55	21	5	0	0
112,5	1	24	86	106	58	14	3	0	0
135	0	21	83	117	68	25	6	1	0
157,5	0	18	78	124	106	48	19	3	1
180	4	30	87	143	161	117	51	13	2
202,5	1	21	79	142	205	174	85	24	6
225	2	25	104	216	273	216	108	31	5
247,5	0	21	111	253	331	211	94	30	7
270	2	34	131	228	246	121	58	17	3
292,5	1	25	98	128	103	54	23	7	2
315	1	29	98	101	73	41	19	5	1
337,5	0	19	82	89	66	37	15	2	0

Table 2.1: Hours per year with a specific wind speed and a specific wind direction.

The average number of hours per year is 8768 hours instead of 8760 because of the impact of the leap years over a series of 30 years.

In order to use this data, in each interval of wind speed, the mean of the range is taken. For example, for the second interval (between 0,3 and 1,6), a wind speed of 0,95 meters per second is taken. So these are the studied data:

Wind speed [m/s]

Wind direction [degrees]

H/YEAR	0,15	0,95	2,5	4,45	6,75	9,4	12,35	15,55	18,05
0	2	30	105	125	100	48	17	3	0
22,5	1	26	97	173	208	107	25	3	1
45	0	21	90	187	178	77	18	3	1
67,5	2	28	108	163	115	45	12	3	0
90	0	18	83	112	55	21	5	0	0
112,5	1	24	86	106	58	14	3	0	0
135	0	21	83	117	68	25	6	1	0
157,5	0	18	78	124	106	48	19	3	1
180	4	30	87	143	161	117	51	13	2
202,5	1	21	79	142	205	174	85	24	6
225	2	25	104	216	273	216	108	31	5
247,5	0	21	111	253	331	211	94	30	7
270	2	34	131	228	246	121	58	17	3
292,5	1	25	98	128	103	54	23	7	2
315	1	29	98	101	73	41	19	5	1
337,5	0	19	82	89	66	37	15	2	0

Table 2.2: Hours per year with a specific wind speed and a specific wind direction.

Given that this wind speed data have been measured at 10 meter height, the altitude correction needs to be considered. The hub height of our wind turbine is 100 meters and the diameter of the blades is 150 meters so, making use of the following equation:

$$v = \frac{v_{ref} \ln(\frac{z}{z_0})}{\ln(\frac{z_{ref}}{z_0})} \quad (2.1)$$

z_0 = **0,00001** for smooth sea and between **0,000015-0,0015** for rough sea,
z_{ref} = 10 meters, the height at which data is measured,
v_{ref} = measured data,
z = the height at which the new speed needs to be calculated,
v = the new wind speed calculated at z height.

With this correction and dividing the area swept by the blades in 4 different parts as shown in the next figure, we get the table 2.3.

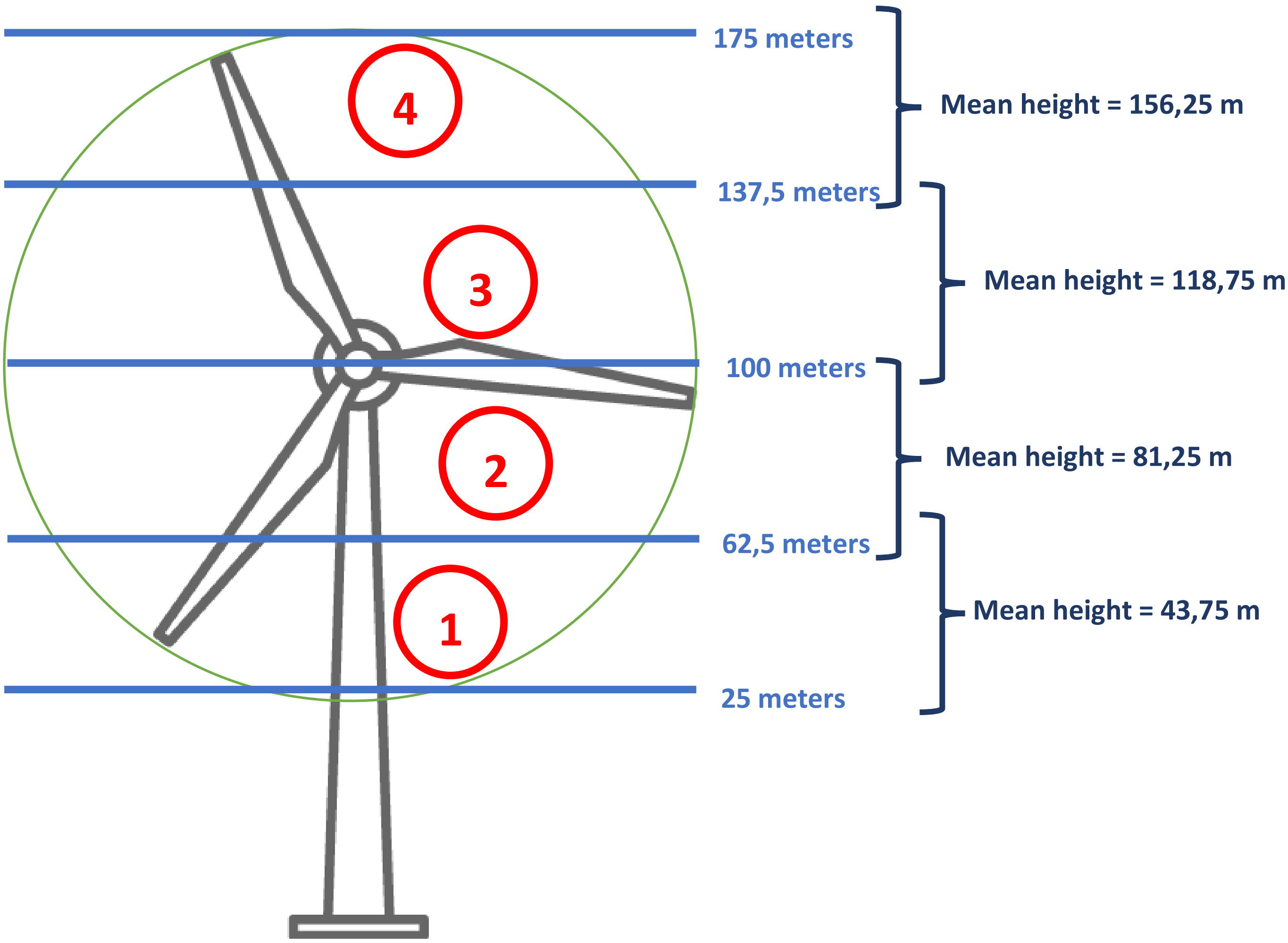


Figure 2.6: Area swept by the blades of a wind turbine.

As said, the area swept by the blades is divided in 4 equal parts: numbers 1, 2, 3 and 4 which are written in red colour in figure 2.6. How much these areas represent on the total of the area swept?

Area_1 = Area_4 and **Area_2 = Area_3**

The area of a circular segment is calculated as follows,

$$A_{segment} = R^2 \cdot \frac{\theta}{2} - \frac{R^2 \cdot \sin(\theta)}{2} = \frac{R^2}{2} \cdot (\theta - \sin(\theta)) \tag{2.2}$$

The total area is $\simeq \mathbf{17671{,}46\ m^2}$,
the area_1 = area_4 $\simeq \mathbf{3454{,}79\ m^2}$ and
the area_2 = area_3 $\simeq \mathbf{5380{,}94\ m^2}$.

So, the area_1 and the area_4 represent each one the **19,55%** of the whole area.
The area_2 and the area_3 represent each other the **30,45%** of the whole area.

Taking a roughness of z0 = 0,00001 (smooth sea) and using equation 2.1:

	Wind speed [m/s]								
	0,15	**0,95**	**2,50**	**4,45**	**6,75**	**9,40**	**12,35**	**15,55**	**18,05**
A1 (43,75m)	0,17	1,05	2,77	4,93	7,47	10,40	13,67	17,21	19,98
A2 (81,25m)	0,17	1,09	2,88	5,12	7,77	10,83	14,22	17,91	20,79
A3 (118,75m)	0,18	1,12	2,95	5,25	7,96	11,08	14,56	18,34	21,28
A4 (156,25m)	0,18	1,14	3,00	5,34	8,09	11,27	14,81	18,64	21,64

Taking a roughness of z0 = 0,0007575 (rough sea average) and using equation 2.1:

	Wind speed [m/s]								
	0,15	**0,95**	**2,50**	**4,45**	**6,75**	**9,40**	**12,35**	**15,55**	**18,05**
A1 (43,75m)	0,17	1,10	2,89	5,14	7,80	10,86	14,27	17,97	20,86
A2 (81,25m)	0,18	1,16	3,05	5,43	8,24	11,48	15,08	18,98	22,04
A3 (118,75m)	0,19	1,20	3,15	5,61	8,51	11,85	15,57	19,61	22,76
A4 (156,25m)	0,19	1,23	3,22	5,74	8,71	12,12	15,93	20,06	23,28

The average of each value in the two tables above and using equation 2.1:

	Wind speed [m/s]								
	0,15	**0,95**	**2,50**	**4,45**	**6,75**	**9,40**	**12,35**	**15,55**	**18,05**
A1 (43,75m)	0,17	1,07	2,83	5,03	7,64	10,63	13,97	17,59	20,42
A2 (81,25m)	0,18	1,13	2,97	5,28	8,01	11,15	14,65	18,45	21,41
A3 (118,75m)	0,18	1,16	3,05	5,43	8,23	11,47	15,07	18,97	22,02
A4 (156,25m)	0,19	1,18	3,11	5,54	8,40	11,70	15,37	19,35	22,46

Table 2.3: Wind speeds recalculated with the height correction depending on the position.

So finally, the wind speed data used for the following analysis is calculated as follows:

Wind speed received at the hub height of a wind turbine in the Parc Éolien en mer de Fécamp = **(wind speed at A1 * 19,55%) + (wind speed at A2 * 30,45 %) + (wind speed at A3 * 30,45%) + (wind speed at A4 * 19,55%)** (2.3)

		Wind speed [m/s]								
	H/YEAR	**0,18**	**1,14**	**3,00**	**5,33**	**8,08**	**11,25**	**14,78**	**18,61**	**21,61**
Wind direction [degrees]	**0**	2	30	105	125	100	48	17	3	0
	22,5	1	26	97	173	208	107	25	3	1
	45	0	21	90	187	178	77	18	3	1
	67,5	2	28	108	163	115	45	12	3	0
	90	0	18	83	112	55	21	5	0	0
	112,5	1	24	86	106	58	14	3	0	0
	135	0	21	83	117	68	25	6	1	0
	157,5	0	18	78	124	106	48	19	3	1
	180	4	30	87	143	161	117	51	13	2
	202,5	1	21	79	142	205	174	85	24	6
	225	2	25	104	216	273	216	108	31	5
	247,5	0	21	111	253	331	211	94	30	7
	270	2	34	131	228	246	121	58	17	3
	292,5	1	25	98	128	103	54	23	7	2
	315	1	29	98	101	73	41	19	5	1
	337,5	0	19	82	89	66	37	15	2	0

Table 2.4: Hours per year with a specific wind speed and a specific wind direction.

2.3. The wind turbine

These are the main features of the Alstom Haliade 150-6MW wind turbine:

Designed in 2012 its unitary electric power is of **6 MW**. This wind turbine can supply power to the equivalent of about 5.000 European homes. Currently, this offshore wind turbine is powering the state of Rhode Island.

The hub (*mât*) height of the wind turbine is **100 meters** and the length of a blade (pale) is **73,5 meters**. So the maximum height of the whole wind turbine is **175 meters** approximately.

Two main features as innovation technology design:

The Pure Torque design protects the generator to ensure and improve its performance by diverting unwanted stresses from the wind through the main frame safely to the turbine's tower.

The innovative Advanced High Density direct-drive offshore wind turbine's Permanent Magnet Generator (PMG) is more compact and lightweight designed compared to earlier generation direct-drive systems.

Each blade is 32.5 tons, the generator (*générateur*) and the gondola (*nacelle*) are 360 tons and the hub is 400 tons more. [4], [10], [9], [11]

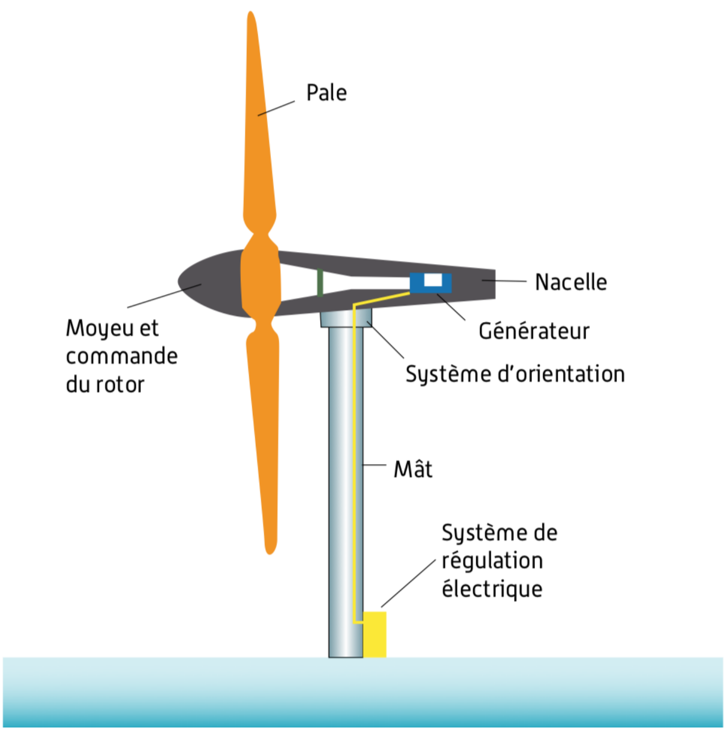


Figure 2.7: Scheme of the Alstom Haliade wind turbine. [4]

Figure 2.8: View of the nacelle of the Alstom Haliade 150-6MW wind turbine. [4]

So, summarizing, the final specifications for the Fécamp wind turbine plant are the following:

Coordinates of the location (x, y)	**49.892, 0.227**
Class	I-B IEC-61400-1 / IEC-61400-3
Type of wind farm	Offshore
Number of wind turbines	83
Area of the wind farm	67 km^2
Expected start up date	2022 year
Expected project lifetime	25 years
Total project capacity	498 MW
Location mean wind speed	9-10 m/s
Weibull C	11,2 m/s
Weibull K	2,26
Cut-in wind speed	3 m/s
Cut-out wind speed	25 m/s
Rotor speed range	4 - 11,5 rpm
Grid frequency	50 / 60 Hz
Wind turbine name	Alstom Haliade 150-6
WT nominal power	6 MW
WT diameter	150 m
WT hub height	100 m
WT weight	857,5 tons

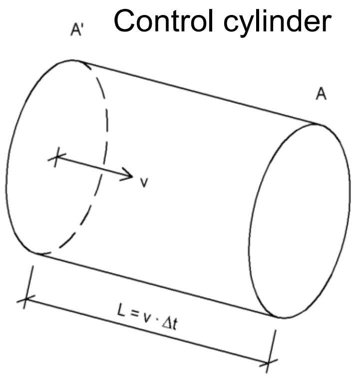
Table 2.5: Specifications for the Fécamp wind turbine plant.

3. Wind turbine aerodynamics

3.1. Wind turbine mechanical system modelling

This chapter analyses the modelling of the dynamics of the mechanical system of a variable speed wind turbine. First of all, a simple aerodynamic model is described. This model relates the power extracted from the wind passing through the surface swept by the turbine with the wind speed and the rotating speed of the turbine. The characteristic parameters for the turbine case that will be used along this project can be found here as well.

Wind turbine generation is based on the extraction of power from the kinetic energy of the wind.



Mass of air in a control cylinder: $m = \rho\, A\, v\, \Delta t$ (3.1)

Then, $dm = \rho\, A\, v\, \mathrm{dt}$ (3.2)

Wind kinetic energy: $E = \frac{1}{2} m\, v^2 = \frac{1}{2} \rho\, A\, v^3\, \Delta t$ (3.3)

$dE = \frac{1}{2} dm\, v^2 = \frac{1}{2} \rho\, A\, v^3\, \mathrm{dt}$ (3.4)

So the wind power is: $P_{wind} = \frac{dE}{dt} = \frac{1}{2} \rho\, A\, v^3$ (3.5)

The power extracted by the turbine (P_t) can be expressed as the kinetic power available in the stream of air across the area swept by its blades (P_{wind}) multiplied by a dimensionless coefficient (C_P) called power coefficient. The C_P can be described as a measure of the aerodynamic efficiency of the turbine and depends on the relation between the average speed of the air across the area covered by the wind turbine blades and its angular speed together with the geometry of the turbine (which also depends on the pitch angle of the blades). The power extracted by the wind turbine has the following expression:

$$P_t = C_p \cdot P_w = C_p \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_w^3 \quad (3.6)$$

where,

ρ is the air density assumed to be constant,

A is the surface covered by the wind turbine blades,

v_w is the average wind speed.

There have been different approaches to model the power coefficient ranging from considering it to be constant for steady state and small signal response simulations to using look-up tables with measured data. A common approach is to use an analytic expression of the form:

$$C_P\,(\Lambda, \beta) = c_1 \cdot \left(c_2 \cdot \frac{1}{\Lambda} - c_3 \cdot \beta - c_4 \cdot \beta^{c_5} - c_6\right) \cdot e^{-c_7 \frac{1}{\Lambda}} \quad (3.7)$$

where [c_1 ... c_9] are characteristic parameters of the wind turbine which can be obtained by statistical analysis of measured data from a real turbine and finite element method simulations, β is the blade pitch angle and Λ is defined as:

$$\frac{1}{\Lambda} \cong \frac{1}{\lambda + c_8 \beta} - \frac{c_9}{1 + \beta^3} \quad (3.8)$$

where λ is the so called tip speed ratio and it is defined as:

$$\lambda \cong \frac{w_t \cdot R}{v_w} \quad (3.9)$$

where ω_t is the turbine speed and R is the turbine radius. To simulate the response of the turbine, the power extracted by the turbine for a given wind speed, angular speed of the turbine and the blade pitch angle, can be easily computed by using (3.6). Dividing the extracted power by the angular speed, the turbine torque, is obtained as:

$$\Gamma_t = C_P \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_w^3 \cdot \frac{1}{w_t} \quad (3.10)$$

The characteristic parameters for a turbine case that will be used along this document can be found in table 3.1. [2], [9]

Parameter	Value	Units	Description
R	75	M	Turbine radius
ρ	1,225	Kg/m³	Air density
c1	0,73		Cp function parameter
c2	151		Cp function parameter
c3	0,58		Cp function parameter
c4	0,002		Cp function parameter
c5	2,14		Cp function parameter
c6	13,2		Cp function parameter
c7	18,4		Cp function parameter
c8	-0,02		Cp function parameter

$c9$	-0,003		Cp function parameter
λ^{opt}	7,206		Optimal tip speed ratio
C_P^{opt}	0,4412		Optimal power coefficient
P_t^N	6	MW	Wind turbine nominal power

Table 3.1: Wind turbine characteristic parameters

The maximum efficiency for a given turbine, can be calculated by differentiating the equation 3.7 as a function of λ and solving the roots of this equation. The pitch angle is taken as zero because an increase on the pitch angle always leads to a reduction on the force that the wind applies on the blades. The maximum efficiency C_P^{opt} becomes:

$$C_P^{opt} = \frac{c_1 c_2}{c_7} \cdot e^{-\frac{c_2 + c_6 c_7}{c_2}} \quad (3.11)$$

also, the optimal tip speed ratio to obtain this efficiency is:

$$\lambda^{opt} = \frac{1}{c_9 + \frac{c_6}{c_2} + \frac{1}{c_7}} \quad (3.12)$$

3.2. Single wind turbine operation

As explained before, wind turbines interact with the wind, capturing part of its kinetic energy and converting it into electrical energy. As the first principle of thermodynamics postulates, this extraction of energy creates a wind energy deficit between the wind leaving the turbine (known as wake) and the wind arriving in front of the turbine. The wind speeds in the rear of the turbines are lower than the upstream wind speeds and, therefore, a reduction of the power extracted is produced at downwind turbines. The turbine wake also causes high turbulence levels in downwind turbines, originating mechanical stress, which may reduce their operating life.

So definitely, the higher the operating point of a wind turbine is, the higher the extraction of energy from the wind is, but the lower the wind speed in the rear of the turbine is.

Thus, in this chapter the operation of a single wind turbine is analysed and the optimal operation point is presented. To carry this out, the following characteristics of a wind turbine presented before are used:

The power P_t generated by a single wind turbine expressed as:

$$P_t = C_p \cdot P_w = C_p \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_w^3 \tag{3.13}$$

The power coefficient C_P written as

$$C_P\,(\Lambda, \beta) = c_1 \cdot \left(c_2 \cdot \frac{1}{\Lambda} - c_3 \cdot \beta - c_4 \cdot \beta^{c_5} - c_6\right) \cdot e^{-c_7 \frac{1}{\Lambda}} \tag{3.14}$$

β, the blade pitch angle and Λ defined as:

$$\frac{1}{\Lambda} \cong \frac{1}{\lambda + c_8 \beta} - \frac{c_9}{1 + \beta^3} \tag{3.15}$$

with λ the so called tip speed ratio defined as:

$$\lambda \cong \frac{w_t \cdot R}{v_w} \tag{3.16}$$

In order to know how much power can be extracted from the wind, it is usual to represent a curve of the generated power versus wind speed, called ideal power curve by using the equation 3.13. The ideal power curve for a typical pitch controlled wind turbine is shown in figure 3.1. It can be observed that the range of operational wind speeds is delimited by the cut-in and cut-out wind speeds. The turbine remains stopped beyond these limits. Below cut-in wind speed, the system would not be profitable since the available wind energy is too low to cover the operational costs. Above cut-out wind speed, the turbine is shut down to protect the wind turbine from damage.

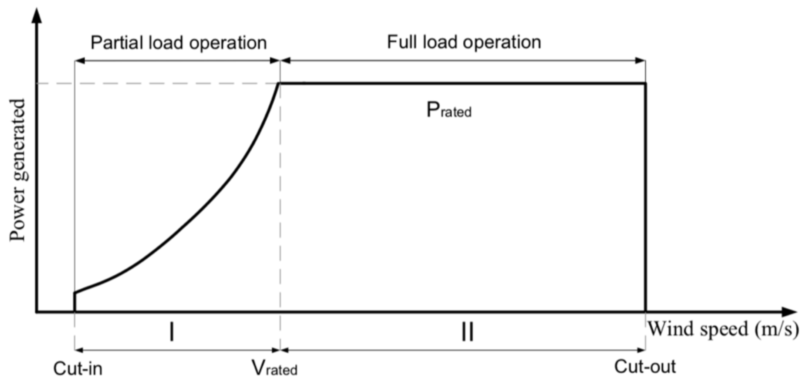


Figure 3.1: Power-wind speed curve of a wind turbine.

There are two different regions with distinctive generation objectives. On the one hand, the objective of the region I, which corresponds to wind speeds lower than the nominal speed, is to extract all the available power from the wind. Accordingly, given a fixed rotor area A, a constant air density ρ and considering the wind speed v_w as an external and uncontrollable variable, the only way to maximise the power Pt of the equation 4.1 is with the power coefficient C_P. Furthermore, in order to maximise the power extracted by a wind turbine, θ_{pitch} is assumed to be zero, so that 3.14 only depends on the tip speed ratio parameter, λ. On the other hand, at high wind speeds (region II), the pitch control is activated with the aim of keeping the power output constant at the nominal power (P_{nom}) in order to avoid overloading.

In figure 3.2, the typical C_P – λ curve is depicted. The C_P – λ curve has a maximum value (C_{max} = C_{opt}) which corresponds to the optimum operating point of the wind turbine (λ_{opt}).

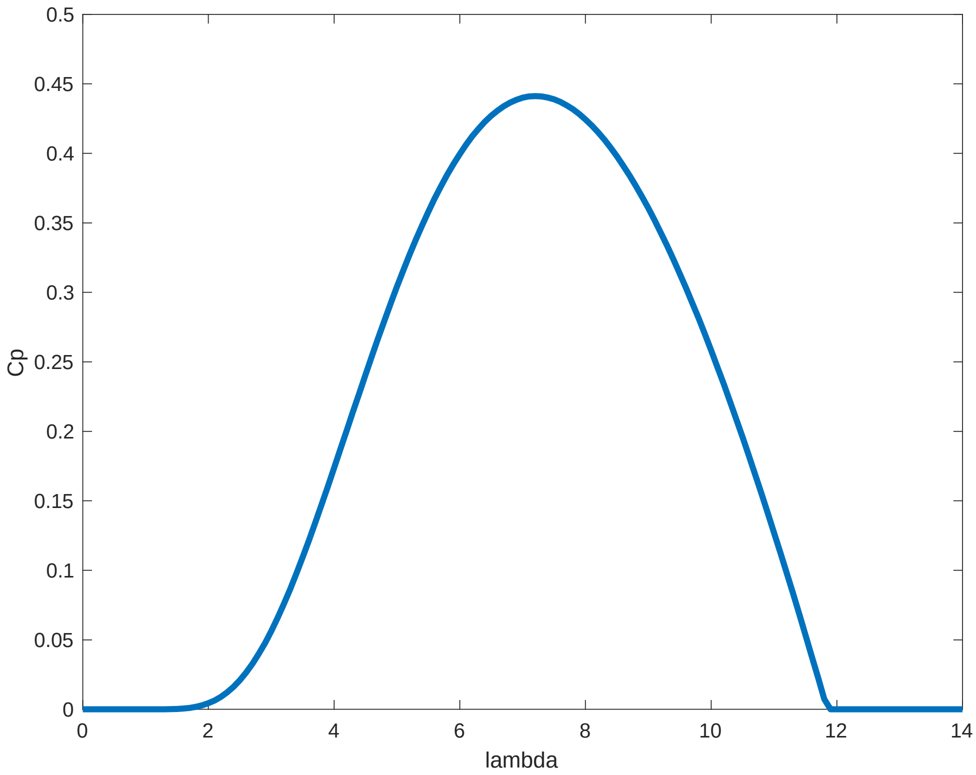


Figure 3.2: Typical C_P – λ curve

With the parameters presented before, this is the Cp - λ curve where the, where the lambda value is **7,206** and the Cp value is **0,4412**, the optimal operating point of one single turbine.

3.3. Wind turbine wake effects

The wake effect is not only dependent on the incoming wind speed and its direction, but also on the design characteristics of the rotor blades, as well as the distance between the turbines. Some studies postulate that wind turbines should be spaced at least 5 to 9 rotor diameters (D) away from each other in the prevailing wind direction and about 3 to 5 rotor diameters for winds coming perpendicularly, in order to reduce the effects of wakes [12]. According to [13] the power losses due to wind turbine wakes if the wind turbines are separated between 4 and 8 rotor diameters from each other result to be in the range of 5 to 10% of the total power generated by the wind power plant. Thus, when the distance between wind turbines increases, the wind speed decay is lower.

Many comprehensive studies have been carried out regarding wind turbine wakes, and several models have been developed by researchers, such as Ainslie's model, Frandsen's model, Mosaic Tile model, Jensen's model [14] and CFD (Computational Fluid Dynamics) model. One of the most widely used wake model, Jensen's model, has been chosen for this study, as it provides adequate accuracy and reduced computational time. It is based on global momentum conservation in the wake downstream of the wind turbine and assumes that the wake downstream of the turbine expands linearly.

The comprehensive wake model implemented for the analysis, considering single, partial and multiple wakes within a wind farm, is detailed below. The study neglects the turbulent behaviour caused by wakes as it does not directly affect the wind power generation output [12].

3.3.1. Single wake

According to Jensen's wake model [14], in the following equation the downstream wind speed of a single turbine is defined (figure 3.3).

$$v_2 = v_1 \cdot \left[1 - \left(\frac{D}{D+2kx}\right)^2 \cdot \left(1 - \sqrt{1 - C_T}\right)\right] \quad (3.17)$$

where v_2 is the wind speed at distance x from the turbine, D is the diameter of the turbine rotor, C_T is the thrust coefficient, v_1 is the free stream wind speed and k is the wake decay constant or opening angle which represents the effects of atmospheric stability. Jensen experimentally found the value of k to be 0,075 for onshore applications and **0,04** for offshore applications. So, in this project the second value of k will be used.

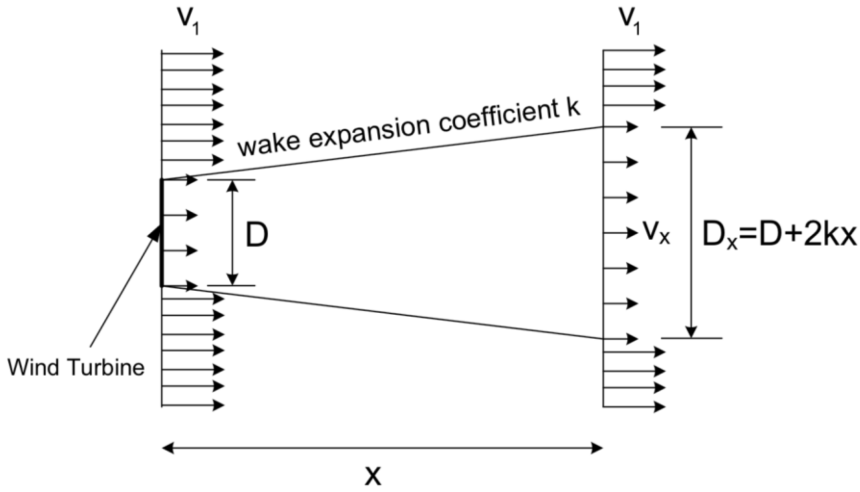


Figure 3.3: Schematic view of the single wake effect.

Generally, the thrust coefficient C_T is wind turbine specific, depending on the blade geometry, the tip speed ratio λ and the applied control strategy of the wind turbine (stall or pitch control). This coefficient can be given by the manufacturer or can be calculated with adequate simulation software and the appropriate simulation.

3.3.2. Partial wake

Partial shadowing occurs when the wake of one or more upwind wind turbines affect partially a downwind turbine, so that the shadow produced by each upwind turbine to the downwind one does not reach the whole area swept by the blades. The wind speed entering into the downwind turbine j affected by the k upwind turbine is then given by [12].

$$v_{Tj} = v_1 \cdot \left(1 - \sqrt{\sum_{k=1}^{N} \beta_{Tj,Tk} \cdot \left(1 - \frac{v_{ps,Tk}}{v_1}\right)^2}\right) \quad (3.18)$$

where v_{Tj} is the wind speed of the downwind turbine j, k is the upwind turbine, v_1 is the initial wind speed entering into the upwind turbine k, $v_{ps,Tk}$ is the shadow of k falling on the jth wind turbine and $\beta_{Tj,Tk}$ is the ratio (the weighting factor) of the shadowing area caused by the wake on the total rotor area. This ratio can be calculated using the following expression:

$$\beta_{Tj,Tk} = \frac{A_{shad}}{A_{wind}} \quad (3.19)$$

with,

$$A_{wind} = \pi \cdot r^2 \quad (3.20)$$

and,

$$A_{shad} = arccos\left(\frac{r_1^2 + d^2 - r_2^2}{2 \cdot r_1 \cdot d}\right) \cdot r_1^2 \; + \; arccos\left(\frac{r_2^2 + d^2 - r_1^2}{2 \cdot r_2 \cdot d}\right) \cdot r_2^2$$
$$- \, sin\left[arccos\left(\frac{r_1^2+d^2-r_2^2}{2 \cdot r_1 \cdot d}\right)\right] \cdot r_1 \cdot d \qquad (3.21)$$

where the parameters r_1, r_2 and d are described in figure 3.4.

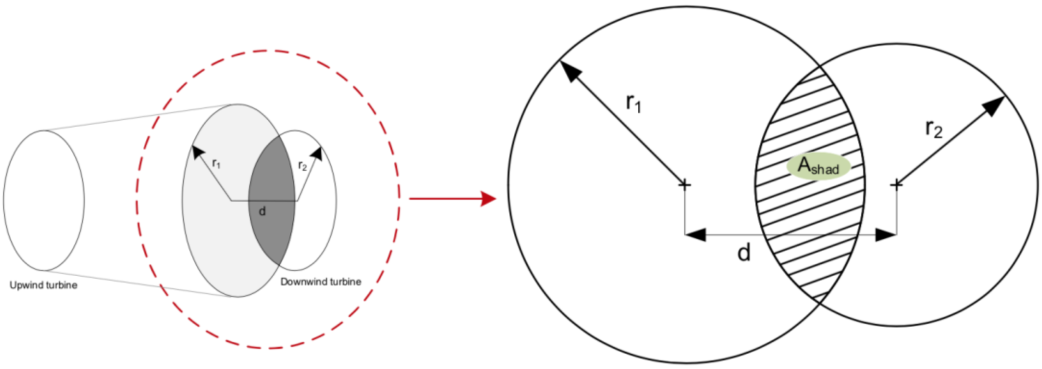


Figure 3.4: Shade area of a downstream wind turbine affected by partial wake.

3.3.3. Multiple wake

In a wind farm with a large number of wind turbines, there are some wind turbines affected by the wakes of several turbines. Therefore, the multiple wake effect should be taken into account. This model assumes that the kinetic energy deficit of interacting wakes is equal to the sum of the energy deficits of the individual wakes. Thus, the speed at the intersection of several wakes is,

$$1 - \frac{v_x}{v_1} = \sqrt{\sum_{i=1}^{N} \left(1 - \frac{v_i}{v_1}\right)^2} \qquad (3.22)$$

where v_1 is the initial free stream speed, N is the total number of upwind influencing turbines, v_i is the wind speed affected by the individual wake i and v_x is the wind speed such that all the wakes are taken into account. [15]

4. Simulation and optimization

Up to date, Wind Power Plants attempt to maximise their power generation by optimising wind turbine performance individually. Besides, the wind plants layouts are also optimised to minimise the wake effect. This fact is especially relevant in offshore where wake effect is more significant than onshore where higher turbulence assists wind speed recovery. Wind turbines are spaced out by a certain distance resulting from a balance between maximising the wind plant energy production by reducing the wake effects and minimising the costs associated with the logistics and electrical interconnections between turbines.

As predicted before, operating each wind turbine at its optimal individual point without considering the impact of the wake effect on the other turbines does not maximise the power output of the whole wind power plant. For this purpose, it is suggested to increase the total wind plant power generated and reduce structural loads by properly operating some wind turbines at non-optimum points, based on the fact that operating the upstream turbines at a lower rotational speed results in higher wind speeds for downstream wind turbines.

In this chapter, a comparative steady state analysis between these two different control strategy approaches is presented. To better understand the optimal wind power plant operation concept, firstly a very simple model consisting of three turbines aligned in a row is considered. Then, a more complex model based on a wind farm composed by 9 wind turbines and taking into account wind directions, is assessed. And finally, the real case of the Parc Éolien en mer de Fécamp is analysed and some conclusions are taken.

Using the known software MATLAB (https://es.mathworks.com) and the steps described in [15], a large procedure has been followed to reach a program or a set of functions able to solve and optimize the power of any wind power plant, defining its variables. The full code implemented is attached in the annexes and it is ready to be used.

4.1. Optimization methodology

4.1.1. 3 Wind turbines in a row case

First of all, a simple example is presented in order to facilitate the comprehension of the proposed methodology. The example aims to validate the hypothesis announced previously that operating some wind turbines at non-optimum points and, therefore allowing the downstream turbines to produce more power can significantly increase the total power produced by the wind power plant, rather than by using the conventional approach based on optimizing the operation of each wind turbine individually. As the figure 4.1 shows, it consists of three wind turbines with a nominal power of **5 MW** and a rotor diameter of **126 m** aligned in a row. The spacing between wind turbines is **7** rotor diameters (D). The free stream speed is chosen as a constant value equal to **9,5 m/s**.

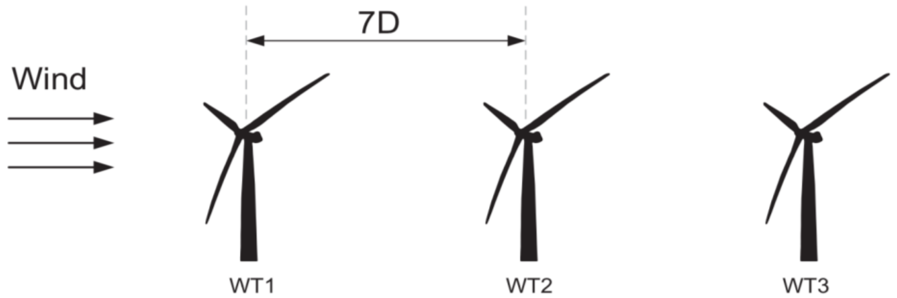


Figure 4.1: Schematic layout of the system under study consisting of three wind turbines aligned in a row.

The power coefficient (C_P) and thrust coefficient (C_T) curves used for the study are shown in figure 4.2.

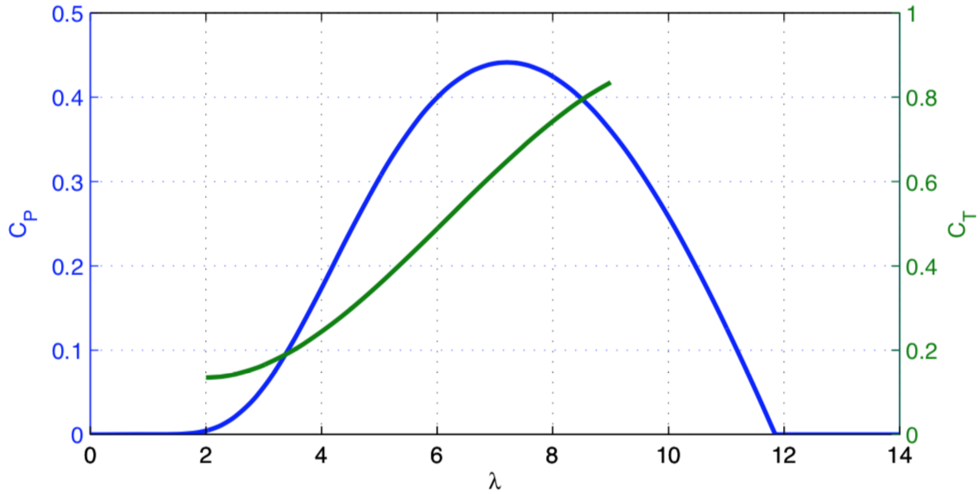


Figure 4.2: Power coefficient (C_P) and thrust coefficient (C_T) used for this simple case.

The procedure of obtaining the optimal operating points of each wind turbine that maximise the total power generation is described below:

- Step 1: First of all, the power generated by the upstream wind turbines is calculated for all their operating points (varying the tip speed ratios, λ_1, from 2 to 9). The power produced by WT1 is computed and can be expressed as

$$P_{WT1} = C_p(\lambda_1) \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_1^3 \qquad \forall\, \lambda_1, \lambda_2 \in [2,9] \qquad (4.1)$$

where v_1 is the upwind speed and the power coefficient, C_P, is only dependent on the tip speed ratio, λ_1, since the pitch angle, θ_{pitch}, is set to zero.

As it is shown, in this case, the optimal tip speed ratio that maximises the power of WT1 is **λ_1 = 7.22**, regardless of the λ_2 value.

With the appropriate code the result in figure 4.3 is obtained (full code in annexes). First of all, the parameters of the first wind turbine are defined, as well as the tip speed ratio and the Cp value equation used. Moreover, the tip speed ratio of the wind turbine 2, λ_2, is also defined to plot the results.

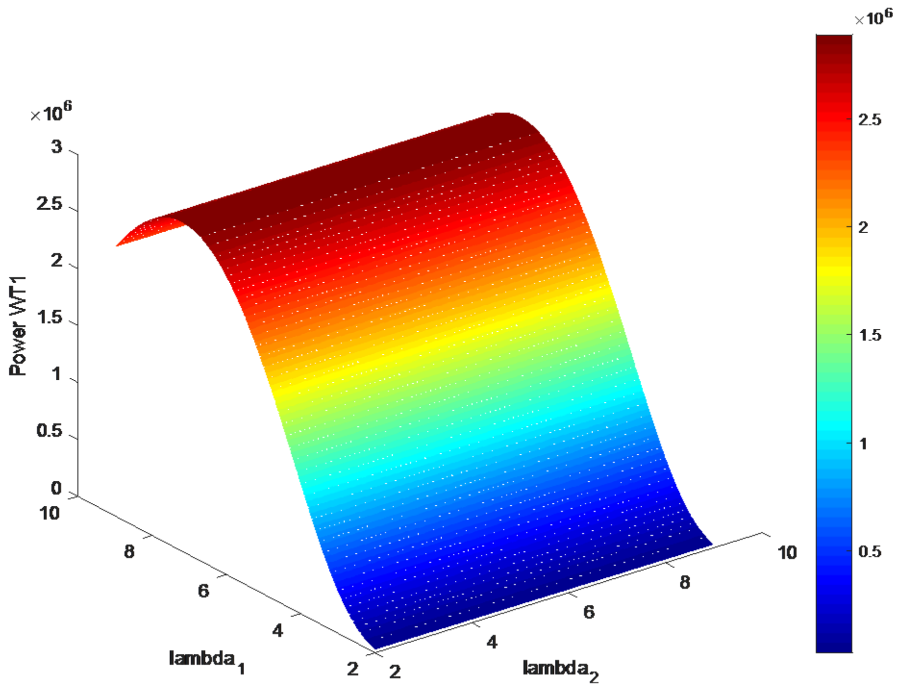


Figure 4.3: Power generated by the upwind turbine (WT1) as a function of λ1 and λ2.

As seen in the previous figure, the power [W] value is independent from the tip speed ratio 2.

- Step 2: Secondly, the power produced by the first turbine affected by the wake effect (in this case WT2) is computed using this equation,

$$P_{WT2} = C_p(\lambda_2) \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_2^3(\lambda_1) \qquad \forall\, \lambda_1, \lambda_2 \in [2,9] \qquad (4.2)$$

As it can be seen, it depends on two parameters: λ_1 and λ_2. The first WT has an influence on wind speed of WT2 (v_2) by modifying the $C_T(\lambda_1)$ value, while the power coefficient $C_P(\lambda_2)$ depends on the second tip speed ratio, similarly to the previous case with WT1. Thus, the resulting surface $P_{WT2}(\lambda_1, \lambda_2)$ of computing (4.2) for all possible combinations of λ_1 and λ_2 parameters, is shown in figure 4.4, through the appropriate code implemented on MATLAB.

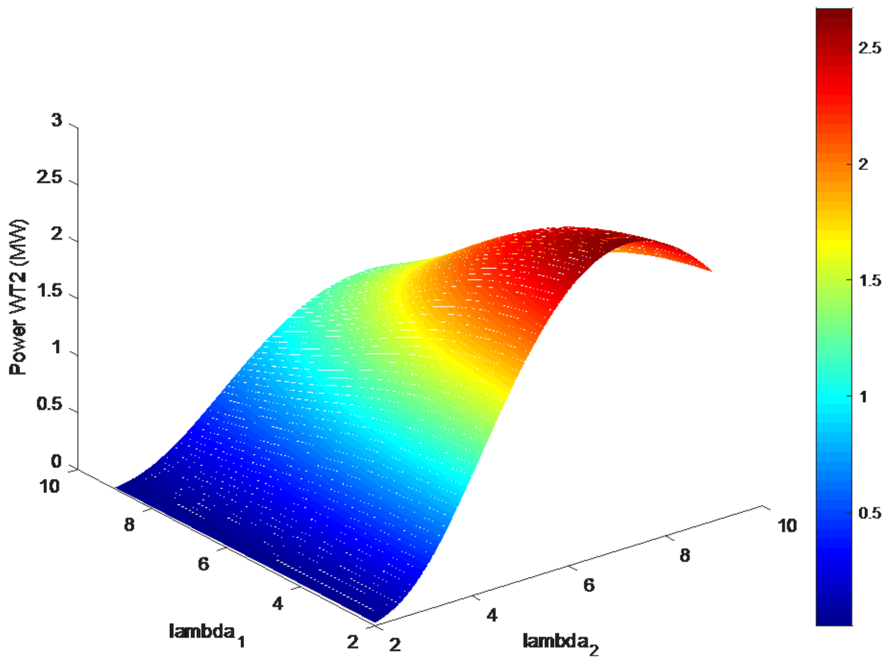


Figure 4.4: Power generated by the WT2 as a function of λ1 and λ2.

As it can be seen in figure 4.4, the maximum power generation for WT2 is achieved when λ_1 is minimum and λ_2 = 7.22. This result is consistent with the fact that the lower the rotational speed of WT1 (lower λ_1), the smaller the impact of the wake effect on downstream wind turbines and, therefore, the greater the power produced by WT2. But, in the end, the value expected to be maximised is the combination of each individual wind turbine power, thereby the sum of them.

- Step 3: The power extracted by WT3 is calculated as

$$P_{WT3} = C_p^{\ max} \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_3^3(\lambda_1, \lambda_2) \qquad \forall\, \lambda_1, \lambda_2 \in [2,9] \qquad (4.3)$$

In this case, the third turbine operates at its optimum point (C_p^{opt}) because no downstream wind turbine is located behind. Regarding its wind speed (v_3), it is computed by the equation of the multiple wakes displayed in 4.13. Figure 4.5 shows the power P_{WT3} obtained by sweeping λ1 and λ2 from 2 to 9.

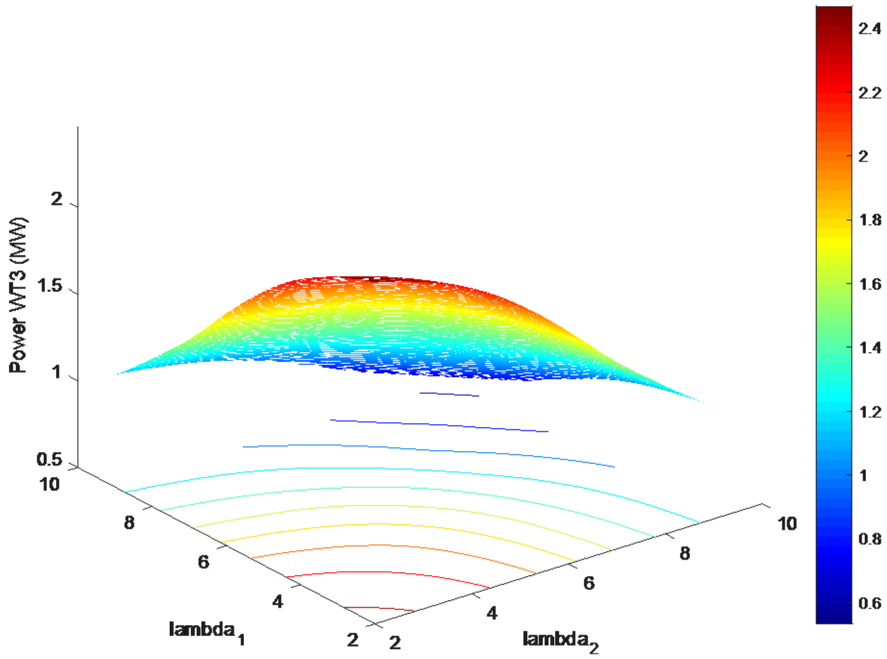


Figure 4.5: Power generated by WT3 as a function of λ1 and λ2.

As expected, the maximum power that can be generated by WT3 occurs when the operating points of WT1 and WT2 are minimum, but again, this is just the power of the third wind turbine and not the whole wind plant power production. The whole power is calculated in the next step, and therefore, the optimal point of the 3 wind turbines will be defined.

- Step 4: Finally, the total power produced by the set of the three wind turbines ($P_{TOT} = P_{WT1} + P_{WT2} + P_{WT3}$) is shown in figure 4.6.

Furthermore, the optimal operating point of each wind turbine is determined considering the maximum value of the total power (the higher point in figure 4.6) and looking up which lambda values for each wind turbine belong to that maximum. Given the new tip speed ratios for each wind turbine, their new optimal operating points can be obtained with the appropriate equation and the results are displayed in the table 4.1.

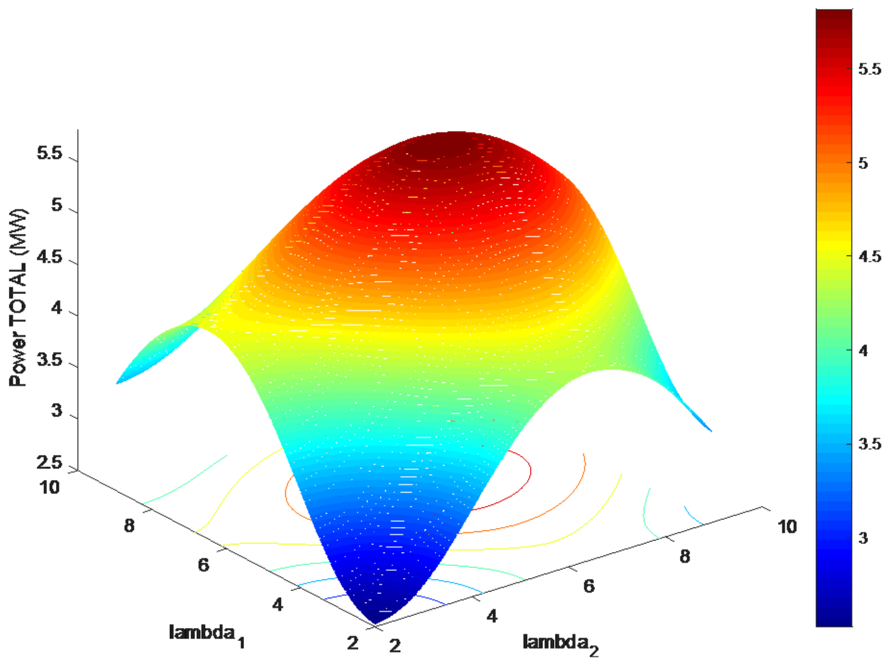


Figure 4.6: Total power generated by the three wind turbines, as a function of λ1 and λ2.

As it can be seen, P_{TOT} reaches its maximum value for $\boldsymbol{\lambda_1}$ **= 6,00** and $\boldsymbol{\lambda_2}$ **= 6,40**. Along all this methodology, the available data of tip speed ratio (λ) for the C_T curve is constrained within the range [2,9] (figure 4.6), but it does not suppose a problem for the purpose of the study since the optimal operation points obtained for each turbine are within these boundaries.

	λ^N	C_P^N
WT1	6,00	0,399
WT2	6,40	0,423
WT3	7,20	0,441

Table 4.1: Optimal operating points of each wind turbine considering wake effect.

In order to compare the operation of each wind turbine for the two mentioned control strategies, figure 4.7 is presented. It shows the tip speed ratio (λ) of each wind turbine and the power generated by each turbine as a function of the free stream wind speed. As it can be seen, WT2 and WT3 reach their nominal power at higher wind speeds when the conventional control strategy is applied because of the increased wake effect.

Furthermore, whereas the three wind turbines operate at their optimum point (λ_{opt}) by considering the conventional control strategy, the proposed control method forces WT1 and WT2 to operate at sub-optimum points. Definitely, the differences can be well appreciated and the proposed control strategy is significantly better.

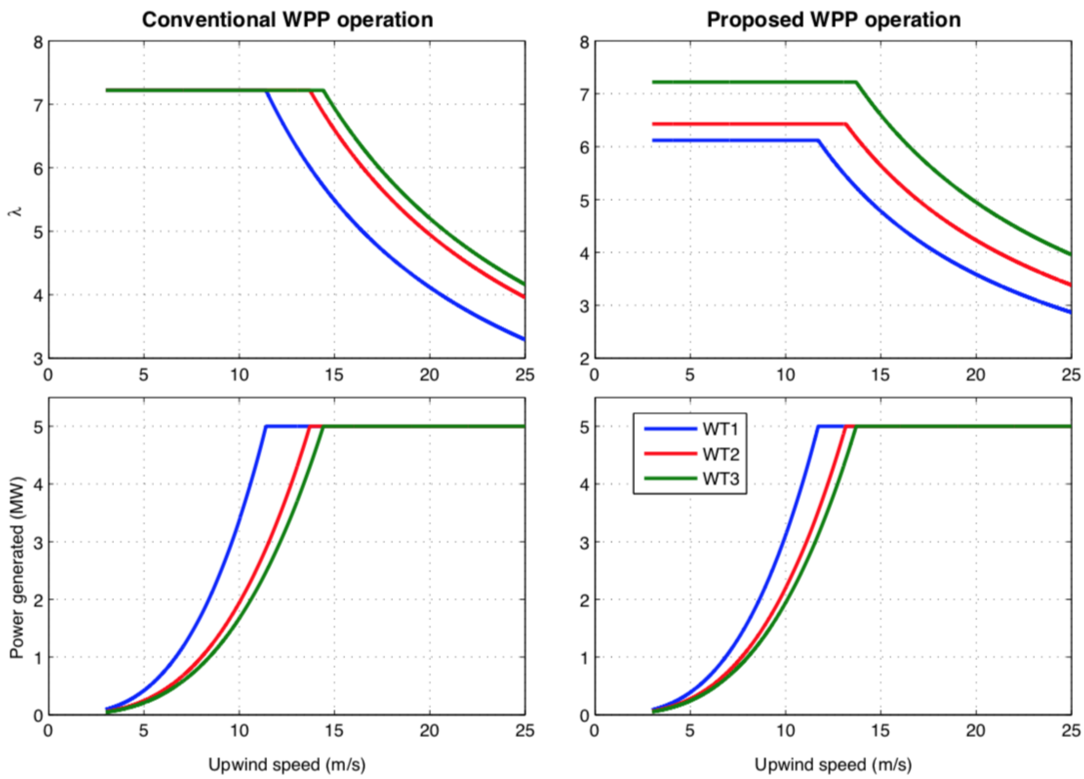


Figure 4.7: Tip speed ratio and power generated by each wind turbine of the study as a function of the wind speed (considering wake effects) for both strategies analysed.

It is important to highlight that the wind direction of the scenario assumed for this conceptual case study is always kept constant (as well as the value, 9,5 meters per second) (best possible scenario for the proposed study). However, in order to accurately quantify both wind power plant operation alternatives, this methodology is applied to a more realistic case study, in which the wind direction is changing with the time, and the layout is more complex.

4.1.2. A 9 Wind turbines in a matrix case

In this chapter, a more complex model based on a wind farm composed by 9 wind turbines and taking into account wind directions, is assessed. This case is implemented to gradually come closer to the reality and to validate again the hypothesis proposed before.

The wind power plant layout of the case under study is shown in figure 4.8. It consists of 9 wind turbines laid out in a rectangular matrix of 3 rows and 3 columns. The spacing between wind turbines is detailed in the figure. Each wind turbine has the same characteristics of the previous case, **5 MW** of nominal power and **126 meters** of rotor diameter.

12 incoming wind direction sectors of 30 degrees each have been considered. With a constant free stream wind speed of **9,5 m/s**.

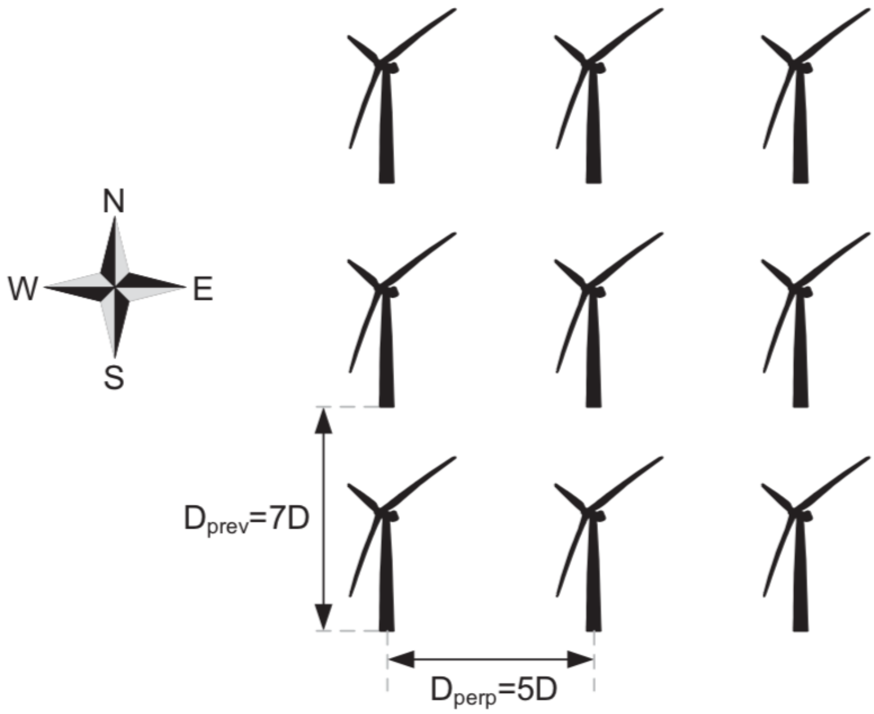


Figure 4.8: Schematic layout of the case under study consisting of 9 wind turbines laid out in a rectangular matrix of 3 rows and 3 columns.

Figure 4.10 shows the wake effect that originates each wind turbine of the wind plant for each wind direction sector considered in the study, and figure 4.9 displays the 30º case individually. As it can be noted, the impact of wake effect on the wind turbines can be classified into three main groups:

- Wind direction sectors of 0º, 90º, 180º and 270º, six wind turbines are completely affected by wakes (three affected by single wakes and three by multiple wakes).
- Wind direction sectors of 30º, 150º, 210º and 330º, four wind turbines are partially affected by wakes (three affected by partial wakes and one by multiple wake).
- Wind direction sectors of 60º, 120º, 240º and 300º, only two wind turbines are partially affected by wakes (both affected by partial wakes and none by multiple wake).

Following the optimization methodology described in the previous section, the procedure of obtaining the optimal operating points of each wind turbine for each wind direction sector that maximise the total power generation is carried out. The code implemented to do so is written in the annexes. Table 4.2 shows the obtained results. It should be noted that wind direction sectors of 0º and 180º are distinguished from 90º and 270º because the spacing between wind turbines is different and therefore, the numerical results change.

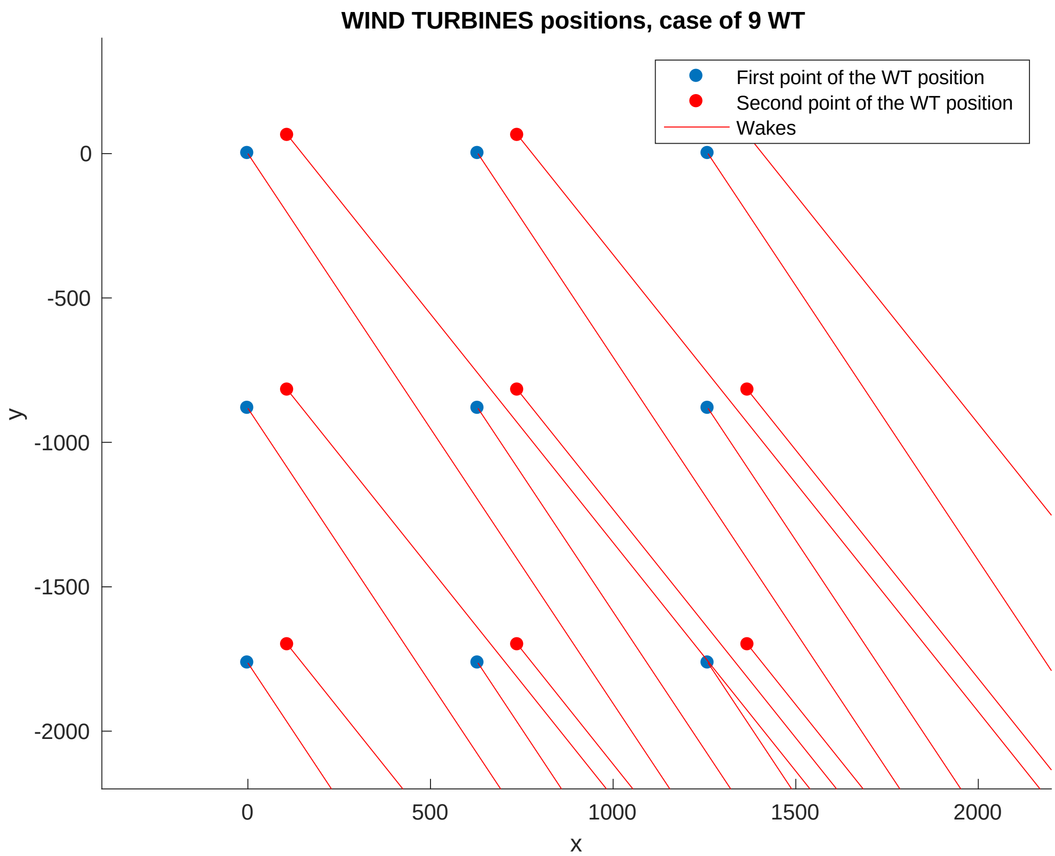


Figure 4.9: Wakes produced by the 9 wind turbines for the case of 30 degrees.

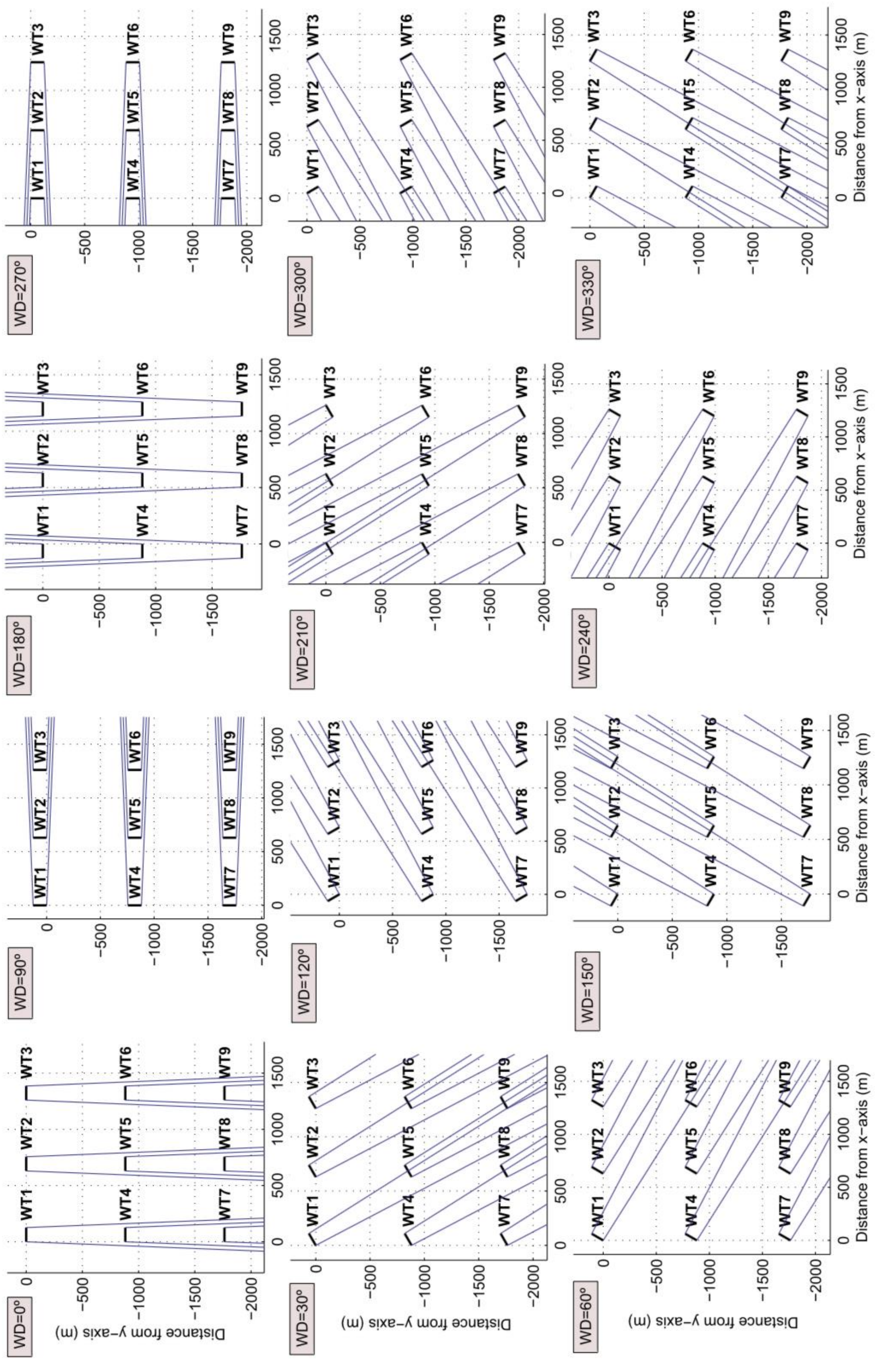


Figure 4.10: Wake effect of each wind turbine of the wind plant for each wind direction sector considered in the study.

(a) Wind directions = 0º and 180º

	λ^N	C_P^N
Upwind turbines	6,01	0,401
WTs affected by single wakes	6,43	0,424
WTs affected by multiple wake	7,21	0,441

(b) Wind directions = 90º and 270º

	λ^N	C_P^N
Upwind turbines	5,90	0,392
WTs affected by single wakes	6,32	0,419
WTs affected by multiple wake	7,21	0,441

(c) Wind directions = 30º, 150º, 210º and 330º

	λ^N	C_P^N
Upwind turbines	6,24	0,415
WTs affected by single wakes	6,24	0,415
WTs affected by multiple wake	7,21	0,441

(d) Wind directions = 60º, 120º, 240º and 300º

	λ^N	C_P^N
Upwind turbines	7,06	0,440
WTs affected by single wakes	7,21	0,441
WTs affected by multiple wake	-	-

Table 4.2: Optimal operating points of all the wind turbines for any wind direction.

Once the nominal operating points of all the wind turbines for any wind direction sector are known, the power generated by each wind turbine as a function of the upwind speed can be determined. It is important to note, for example, that in the table 4.2, the case (d) is barely affected by the wake effect, as these directions create wakes that have reduced impact on the downwind turbines. Therefore, the difference between the power generated with the conventional wind power plant control and the proposed one is very low, as seen in figure 4.14.

In the following figures, the difference between the power generated with both wind power plant control strategies is highlighted.

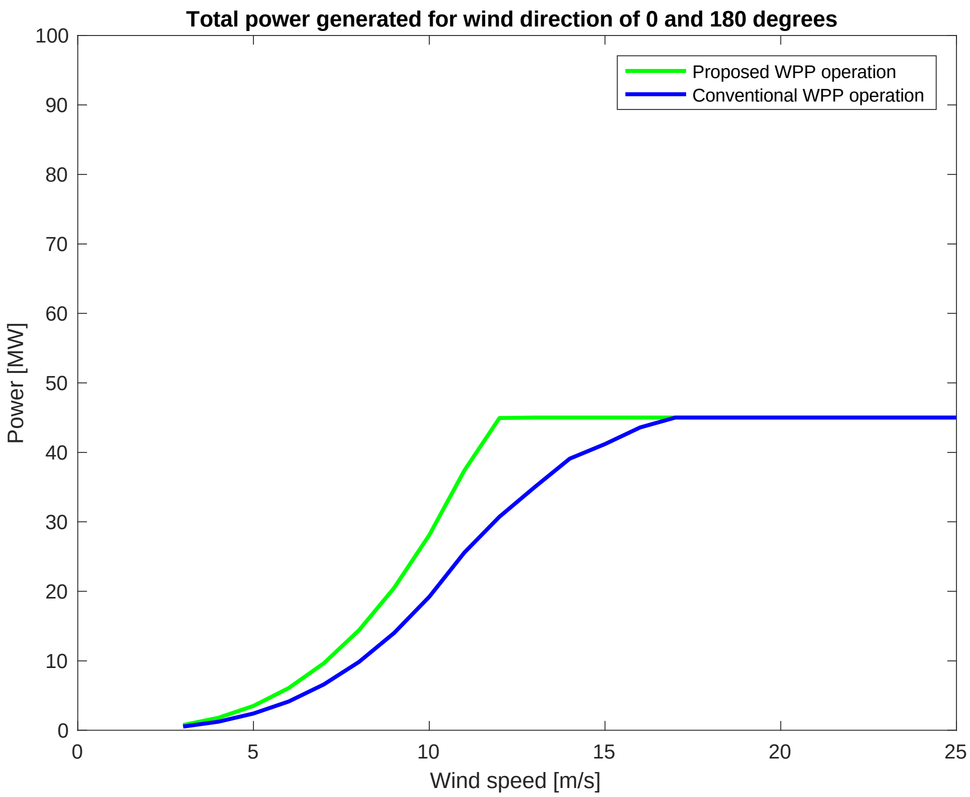


Figure 4.11: Power generated of the whole wind plant, for 0 and 180 degrees as wind direction, vs the wind speed in m/s.

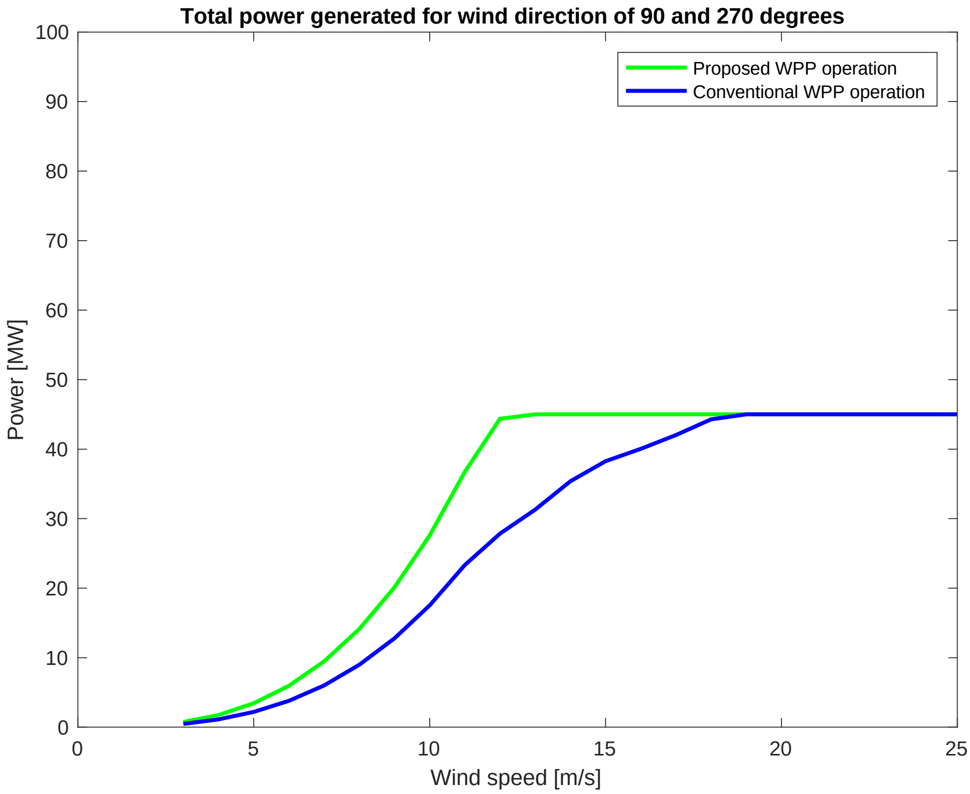


Figure 4.12: Power generated of the whole wind plant, for 90 and 270 degrees as wind direction, vs the wind speed in m/s.

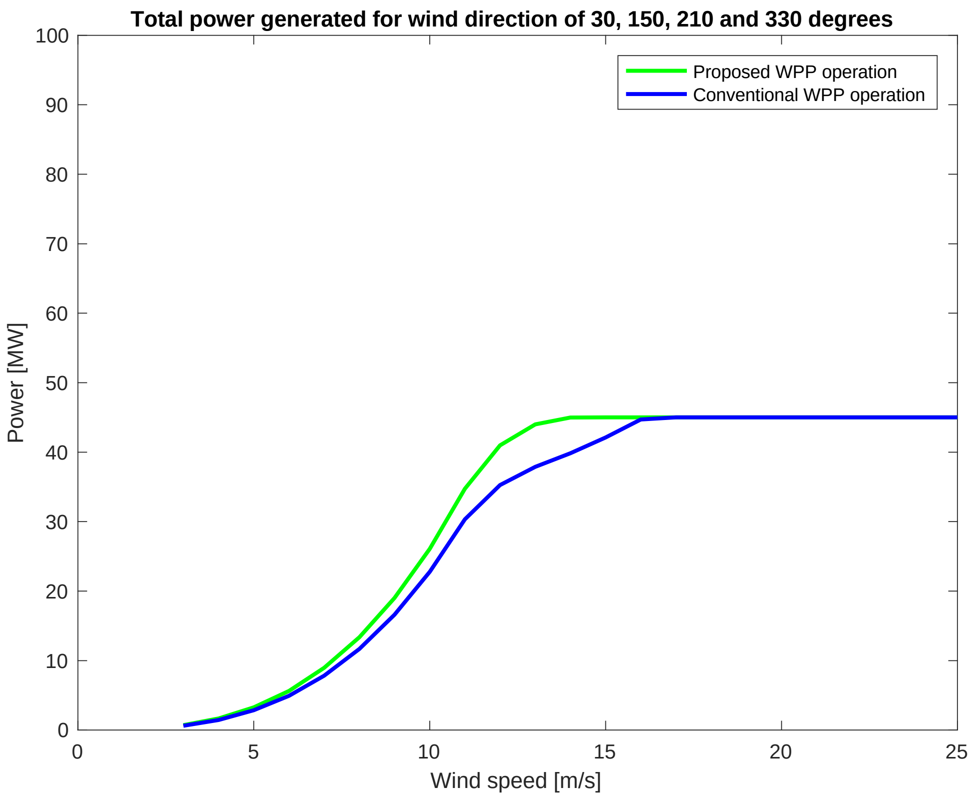


Figure 4.13: Power generated of the whole wind plant, for 30º, 150º, 210º and 330º as wind direction, vs the wind speed in m/s.

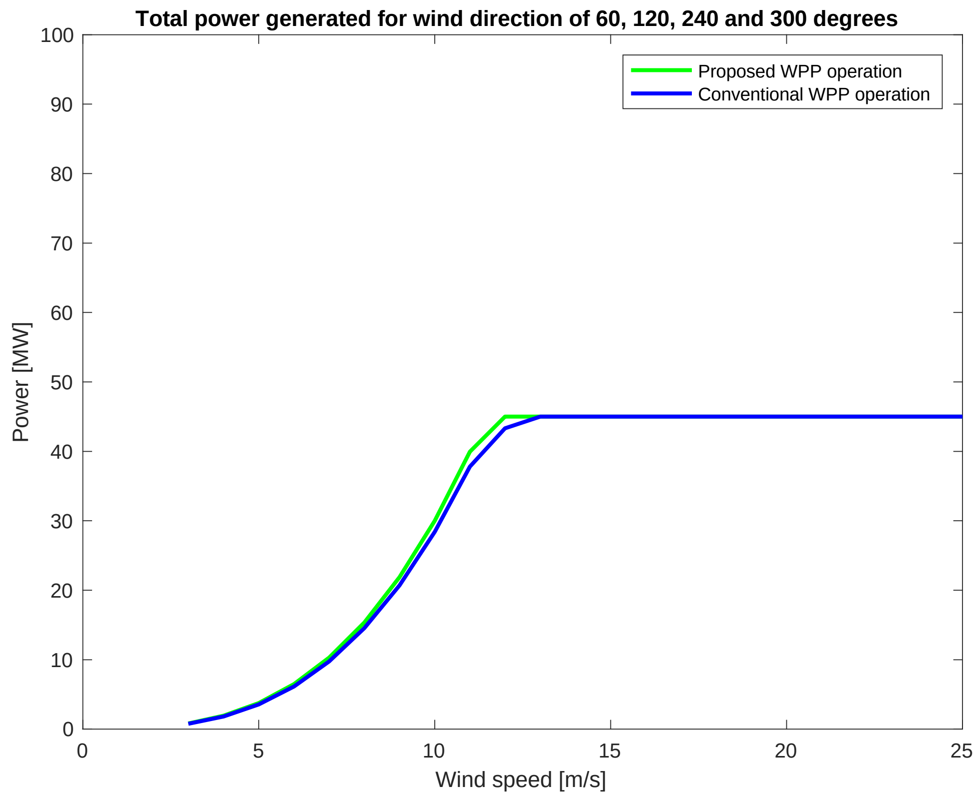


Figure 4.14: Power generated of the whole wind plant, for 60º, 120º, 240º and 300º as wind direction, vs the wind speed in m/s.

Definitely, the proposed wind power plant operation is more efficient in total power generation terms. In all the 4 cases, the power generated by the proposed strategy is higher along all the different wind speeds and equal to the other strategy when the nominal total power is reached.

4.2. Application case: "Le Parc Éolien en mer de Fécamp"

Until here, ideal wind turbines dispositions have been analysed in order to develop an optimization methodology able to be written in MATLAB script to validate the hypothesis of the project. In this section, a generic code has been developed, with the help of many MATLAB functions, to reach a program able to calculate the power of a whole wind power plant, and moreover, to calculate the wind speed that arrives to each wind turbine taking into account the wake effect of all the other turbines of the wind plant. The characteristics of the wind turbine, the disposition of each wind turbine in the wind plant, and the wind characteristics of the zone (wind speed and wind direction) are required to be defined previously to run the code.

Now it is time to transfer all this scripts and formulas to the playground of this project, Le Parc Éolien en mer de Fécamp. The characteristics of the wind plant and the wind turbine used in this section have been described before (see section 2), in order to simulate as better as possible the response of this wind farm.

To start with the program implementation, a free wind speed constant and equal for all the cases (9,5 m/s) has been used. Later, the wind speed will be analysed and the realistic wind rose, described in section 2 as well, will be used.

To get this generic script, able to be adapted to any wind plant, the optimal operation point for all the wind turbines, the Cp parameter in his optimal value and the corresponding Ct have been considered, specifically this optimal point is determined with a tip speed ratio of **7,21** and a Cp of **0,441**. So, in this case, the conventional wind power plant operation is implemented. To implement the proposed optimization methodology explained before is far more complex due to the fact that there are 83 wind turbines and one turbine can be affected by the wake of many other, this fact increases a lot the computational time and, thus, the difficulty. In any case, many results and conclusions will be exposed using the conventional operation strategy.

The MATLAB program attached to the annexes is able to be adapted to any wind plant with just defining the characteristics of the wind turbine (the blades length, hub height, the operation point [the tip speed ratio or the Cp parameter] and the cut-in and cut-out wind speed), the characteristics of the wind (air density, the wind speed and the wind direction [or an specific wind rose data]) and finally the characteristics of the wind power plant (the number of operating wind turbines and the layout [x, y position of each wind turbine]). With just this input data, the power of the whole plant, the power generated by each turbine and the wind speed under wake effects arriving to each turbine is calculated and displayed in some different plots.

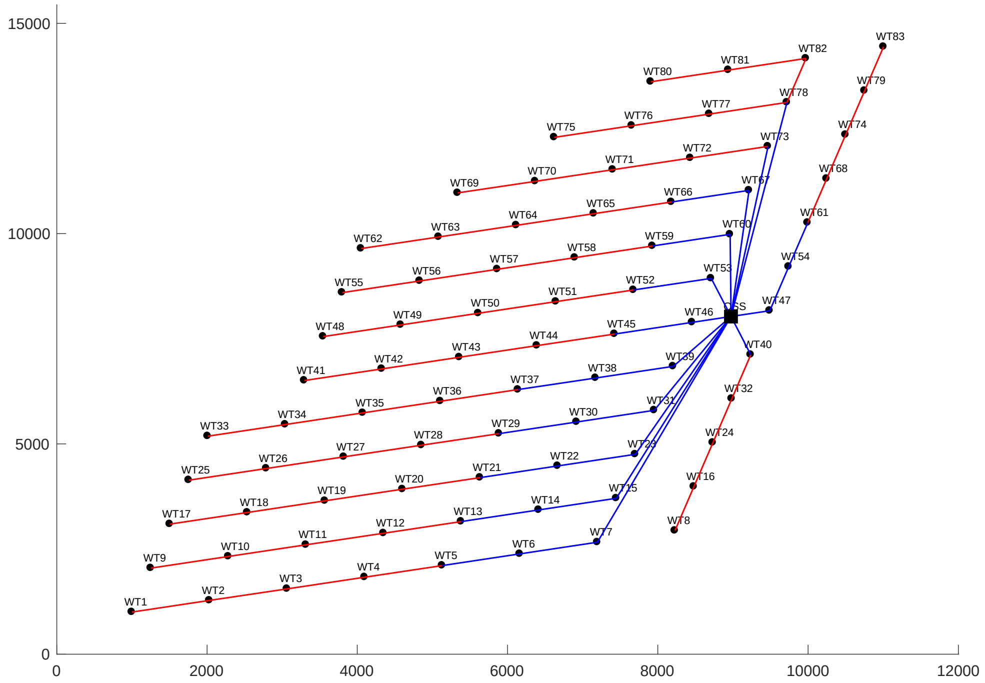


Figure 4.14: “fecamp layout.fig”

First of all, the wind turbines locations are defined using the figure “fecamp layout.fig” to extract the exact positions as numeric values, (X Y) plot positions.

To do so an [83x2] matrix is defined, where each row is a wind turbine and the first column is the position X and the second, the Y position, of the first point of a wind turbine. In figure 4.14 this first point is displayed.

Just after, the second point of each wind turbine is defined (see figure 4.15). Note that this second point depends on the value of alpha, where alpha is the inclination of the wind turbine disposition regarding the wind direction, because the wind turbine rotates in order to keep the perpendicularity against the wind direction (its maximum blades performance) (see figure 4.15 as well).

Hence, the figure 4.15 also displays the wake effect casted by each wind turbine. As it can be seen, it depends on the alpha value, so the wake lines rotate while the wind turbines face the wind direction.

Thus, the wind turbines points (1 and 2) and the wind turbines wake lines are plotted and the following figure is obtained, which is necessary to determine the wake effect that receive each wind turbine.

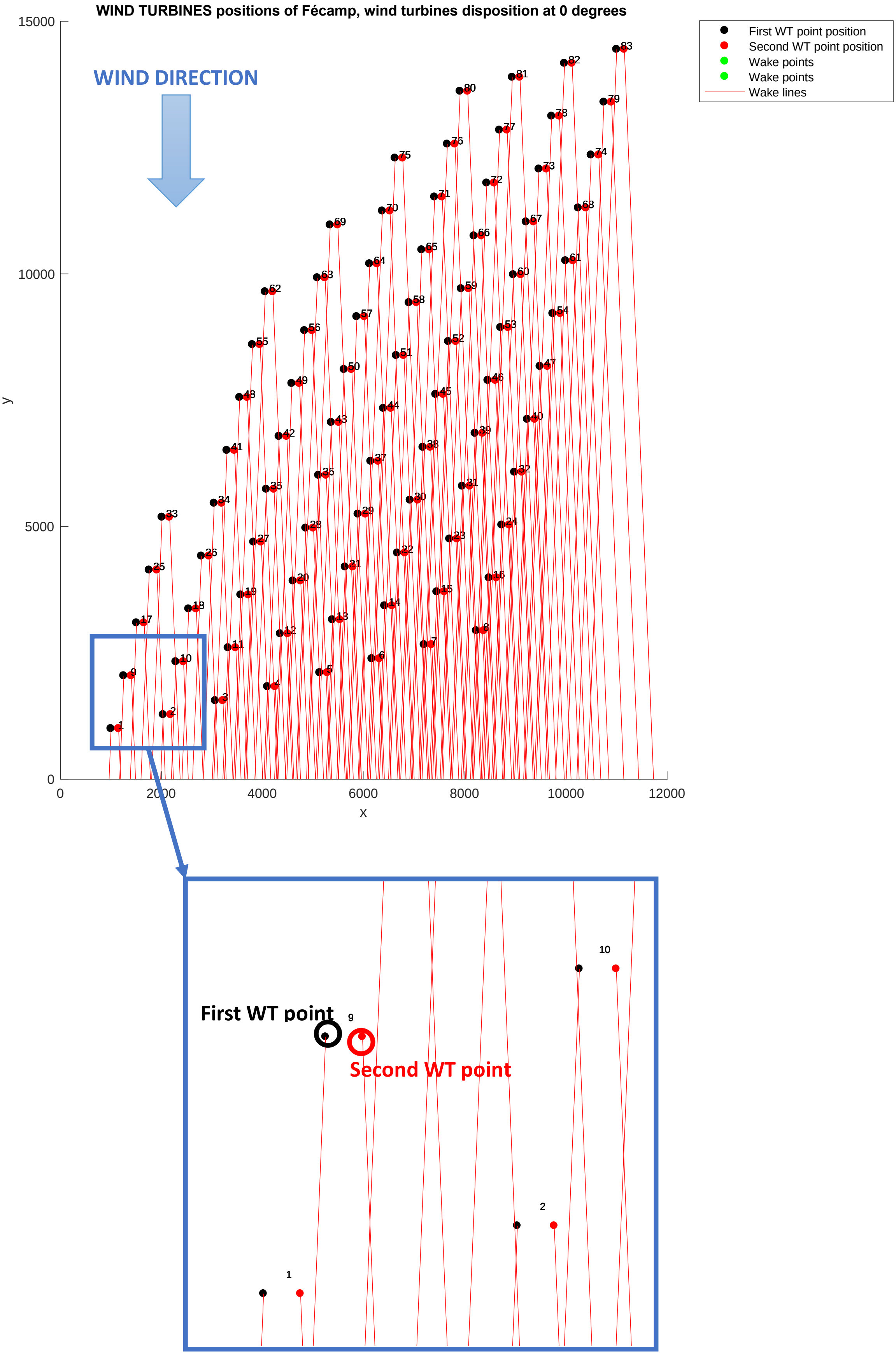


Figure 4.15: Fécamp layout and wind turbines corresponding wake lines

The following figure 4.16 displays the alpha angle, because the figure 4.15 shown before is the Fécamp layout with an alpha value of 0 degrees and it is difficult to appreciate the angle variation.

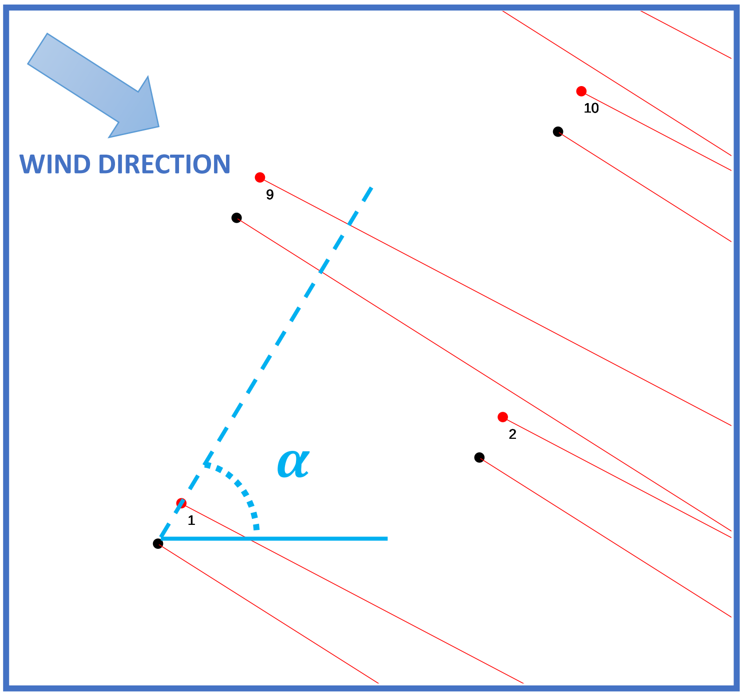


Figure 4.16: Angle alpha representation.

Once the wind turbines layout is achieved, the next step is to obtain a matrix with as many rows and columns as wind turbines has the plant, in this case an [83x83] matrix. So if "n" is the row number and "m" the column number, the position [n,m] tells that the wind turbine number "n" receives Single wake, Partial wake or nothing from the wind turbine number "m". Thus, each cell of this matrix contains "Single wake", "Partial wake" or it is empty.

This matrix has been filled following the theory described in the section 4.2, the wake model developed by Jensen. Using MATLAB, a code is implemented in order to obtain the matrix. This code is quite large because it is written in such a way that can be used with any wind power plant layout. The methodology used to determine if a wind turbine receives Single, Partial, Multiple wake or nothing is the following:

- If neither the point 1 nor the point 2 of a wind turbine is inside the area defined by the wake lines of another wind turbine, the cell is kept empty.
- If both points (point 1 and point 2) of a wind turbine are inside the area defined by the wake lines of another wind turbine, the cell is filled with "Single wake".
- If either the point 1 or the point 2 of a wind turbine is inside the area defined by

another wind turbine, the cell is filled with “Partial wake”.

- If a row “n” has more than one filled cell, this wind turbine “n” is receiving multiple wakes from different wind turbines.

To make it easier to figure out how this matrix is, the following table (4.3) shows a part of it, defining the angle alpha as 0 degrees.

WTs 1-74

WTs 21-83

WT	...75	76	77	78	79	80	81	82	83
1									
2									
3									
4									
5									
6	partial wake								
7		partial wake							
8			partial wake			single wake			
9									
10									
11									
12									
13									
14	single wake								
15		single wake				partial wake			
16			single wake				partial wake		
17									
19									
20...									

Table 4.3: Wind turbine wakes matrix with a wind direction of 0 degrees.

For example, the cell (6, 75) tells that the wind turbine 6 is affected by the wakes of the wind turbine 75, in particular, partial wake.

Once this matrix is defined, it is time to calculate the wind speed that reaches each wind turbine. Once again, the theory described in the section 4.2 is used to determine the wind speed affected by the different types of wakes. Hence, the objective now is to obtain a [83x1] vector, where the row “n” has the wind speed that reaches the wind turbine “n”, in m/s.

4 different loops are written to cover all the different wake cases:

1. The first loop assigns the free stream velocity to those wind turbines that are not affected by any wake, empty rows.
2. The second loop calculates the wind speed that reaches those turbines which are only affected by single wakes.

3. The third loop calculates the wind speed that reaches those turbines which are only affected by partial wakes.
4. The forth loop calculates the wind speed that reaches those turbines affected by one or more than one single wakes and one or more than one partial wake at the same time.

So the wind speeds that reach all the wind turbines in the plant are calculated and stored in the vector. Finally, the power of the whole wind power plant is calculated and the different plots are shown below.

By default, the power of the whole wind plant is obtained for a specific wind direction (value of angle alpha and free stream wind speed to be defined when calling the function), and taking into account the parameters of the Parc Éolien en mer de Fécamp: the radius of the blades (**75 meters**), the nominal power of the wind turbine (**6 MW**) and the cut-in and cut-out wind speed of the wind turbines (**3 and 25 meters per second**). Next, different wind conditions will be evaluated and the profitability of the whole Fécamp project will be analysed and discussed.

First of all, here is a distribution of the total power extracted from the plant with a free stream wind speed of 9,5 meters per second (the average wind speed in the Fécamp location) and along the 360 degrees that the wind direction can take.

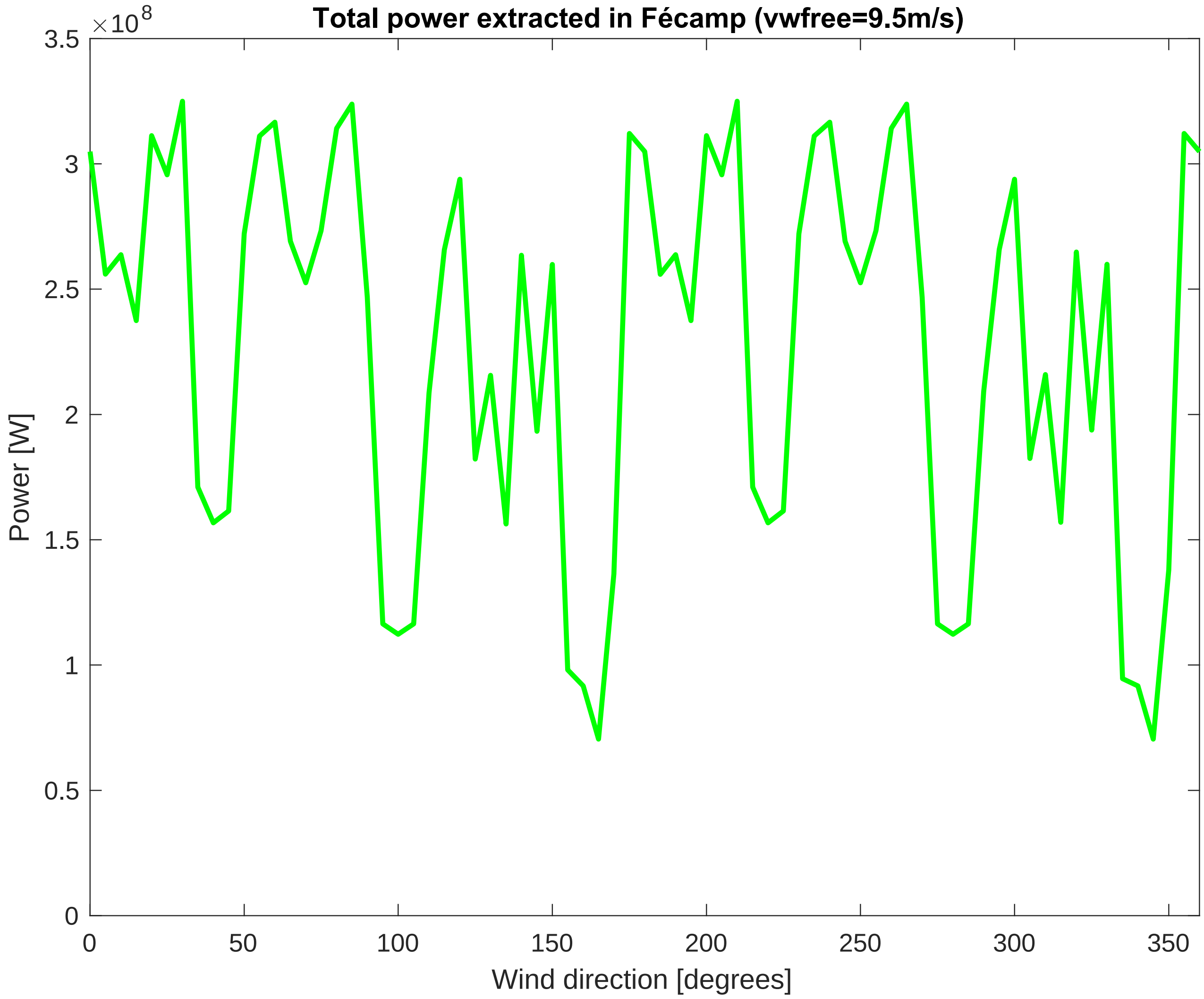


Figure 4.17: Total power generated vs wind direction in Parc Éolien en mer de Fécamp.

As seen in this figure, there are some wind directions much better than others in terms of power generation due to the fact that, the layout of the wind power plant has been designed regarding the most common wind directions for that specific area in order to avoid the wake effect as much as possible.

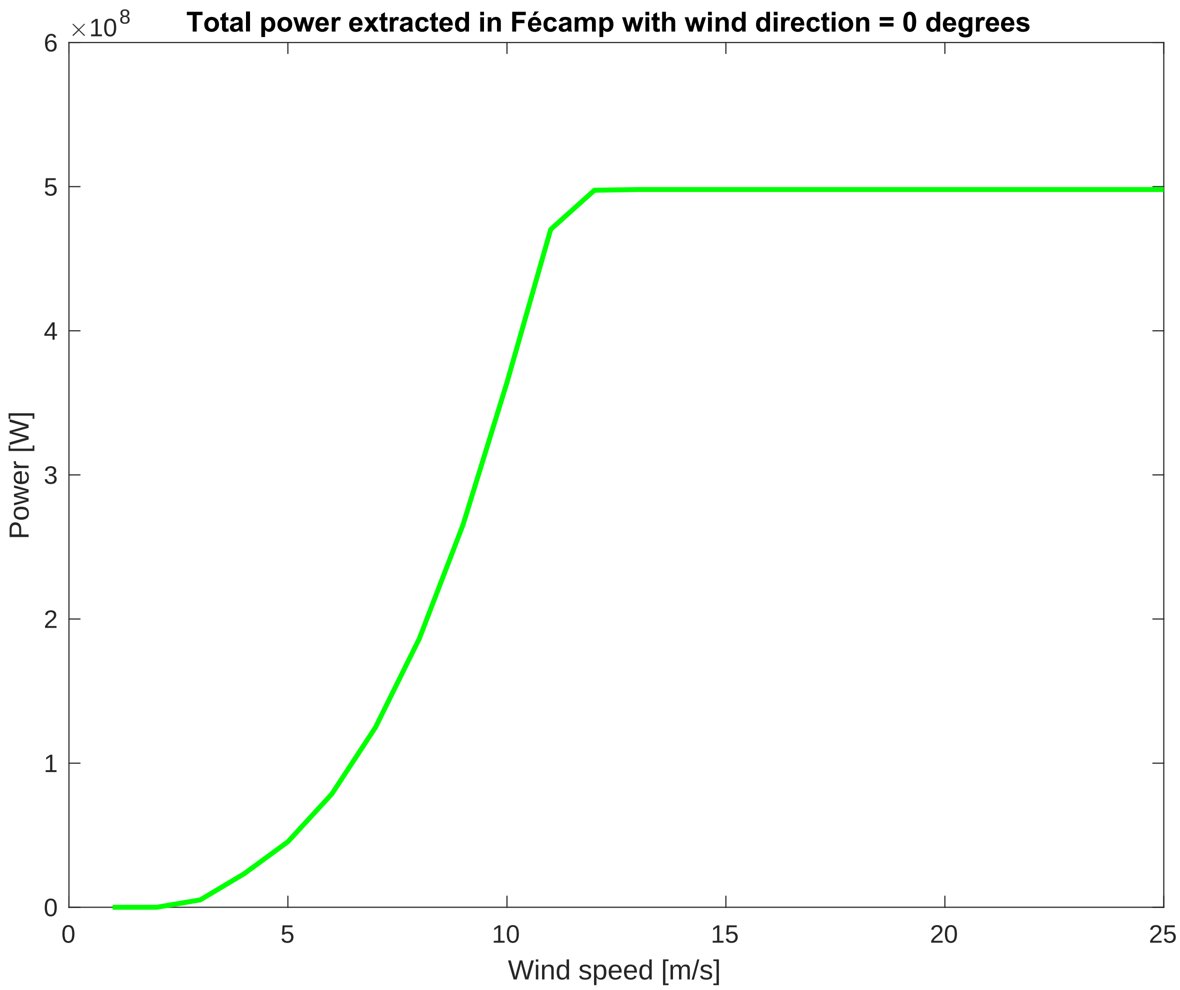


Figure 4.18: Total power generated vs wind speed in Parc Éolien en mer de Fécamp.

This second figure is useful because it shows that with a wind speed above 12,5 m/s approximately the wind power generated is kept constant. This is due to the fact that the nominal power of these wind turbines is 6 MW, so when it is surpassed the turbine is set to extract as maximum as the nominal power. This power regulation is carried out through the angle pitch control, as explained in the section 3.2.

Furthermore, the figure below results as a combination of figure 4.17 and 4.18. It shows the power generated in the Parc Éolien en mer de Fécamp per wind direction and wind speed. 9 different values of wind speed are chosen (0,18 1,14 3,00 5,33 8,08 11,25 14,78 18,61 21,61) [m/s]. The wind direction is divided in 16 different values (0 22,5 45 67,5 90 112,5 135 157,5 180 202,5 225 247,5 270 292,5 315 337,5) [degrees].

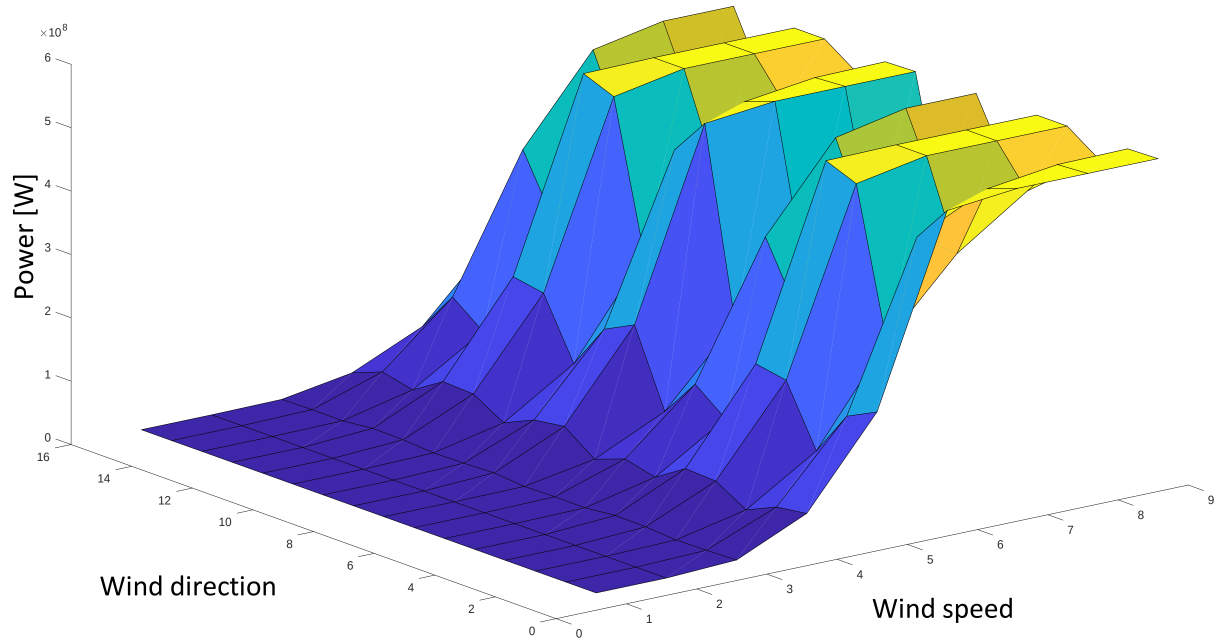


Figure 4.19: Power produced per wind direction and wind speed in Parc Éolien en mer de Fécamp.

Taking the table 2.4 from the section 2.2, as it is the wind distribution in the Parc Éolien en mer de Fécamp, split by the wind direction [degrees] and the wind speed [m/s] that affect the swept area by the blades of the wind turbines, the total energy extracted by the wind power plant during a whole year can be calculated.

Wind speed [m/s] / Wind direction [degrees]

H/YEAR	0,18	1,14	3,00	5,33	8,08	11,25	14,78	18,61	21,61
0	2	30	105	125	100	48	17	3	0
22,5	1	26	97	173	208	107	25	3	1
45	0	21	90	187	178	77	18	3	1
67,5	2	28	108	163	115	45	12	3	0
90	0	18	83	112	55	21	5	0	0
112,5	1	24	86	106	58	14	3	0	0
135	0	21	83	117	68	25	6	1	0
157,5	0	18	78	124	106	48	19	3	1
180	4	30	87	143	161	117	51	13	2
202,5	1	21	79	142	205	174	85	24	6
225	2	25	104	216	273	216	108	31	5
247,5	0	21	111	253	331	211	94	30	7
270	2	34	131	228	246	121	58	17	3
292,5	1	25	98	128	103	54	23	7	2
315	1	29	98	101	73	41	19	5	1
337,5	0	19	82	89	66	37	15	2	0

Table 2.4: Hours per year with a specific wind speed and a specific wind direction.

Wind speed [m/s] / Wind direction [degrees]

P[W]	0,18	1,14	3,00	5,33	8,08	11,25	14,78	18,61	21,61
0	0	0	5,16E+10	5,51E+11	1,92E+12	4,86E+12	4,98E+08	4,98E+08	4,98E+08
22,5	0	0	3,35E+10	4,81E+11	1,68E+12	4,27E+11	4,80E+11	4,95E+12	4,96E+11
45	0	0	2,45E+10	2,65E+11	9,62E+11	2,45E+12	3,61E+12	4,36E+12	4,65E+12
67,5	0	0	5,93E+10	5,47E+11	1,90E+12	4,77E+12	4,98E+08	4,98E+08	4,98E+08
90	0	0	6,70E+10	5,72E+11	1,99E+12	4,96E+12	4,98E+08	4,98E+08	4,98E+08
112,5	0	0	1,55E+10	2,65E+11	9,51E+11	2,47E+12	3,54E+12	3,84E+12	4,01E+11
135	0	0	1,93E+10	3,81E+11	1,33E+12	3,42E+12	4,75E+12	4,98E+12	4,98E+08
157,5	0	0	1,03E+10	2,04E+11	7,24E+11	1,89E+12	2,36E+12	2,55E+11	2,64E+12
180	0	0	5,16E+10	5,51E+11	1,92E+12	4,86E+12	4,98E+08	4,98E+08	4,98E+08
202,5	0	0	3,35E+10	4,81E+11	1,68E+12	4,27E+11	4,80E+11	4,95E+12	4,96E+11
225	0	0	2,45E+10	2,65E+11	9,62E+11	2,45E+12	3,61E+12	4,36E+12	4,65E+12
247,5	0	0	5,93E+10	5,47E+11	1,90E+12	4,77E+12	4,98E+08	4,98E+08	4,98E+08
270	0	0	6,70E+10	5,72E+11	1,99E+12	4,96E+12	4,98E+08	4,98E+08	4,98E+08
292,5	0	0	1,55E+10	2,69E+11	9,58E+11	2,49E+12	3,58E+12	3,80E+12	3,94E+08
315	0	0	1,93E+10	3,81E+11	1,33E+12	3,42E+12	4,76E+11	4,98E+08	4,98E+08
337,5	0	0	9,03E+09	1,94E+10	6,89E+11	1,80E+12	2,41E+11	2,70E+12	2,81E+11

Table 4.4: Power produced with a specific wind speed and a specific wind direction.

With the power and the hours per year for each wind speed and wind direction, and knowing this relation **Energy = Power * time**, the next table is obtained:

E[WH]	Wind speed [m/s] 0,18	1,14	3,00	5,33	8,08	11,25	14,78	18,61	21,61
Wind direction [degrees]									
0	0	0	5,42E+12	6,89E+13	1,92E+14	2,33E+14	8,47E+09	1,49E+09	0,00E+00
22,5	0	0	3,25E+12	8,33E+13	3,49E+14	4,57E+13	1,20E+13	1,48E+13	4,96E+11
45	0	0	2,20E+12	4,95E+13	1,71E+14	1,89E+14	6,49E+13	1,31E+13	4,65E+12
67,5	0	0	6,41E+12	8,91E+13	2,19E+14	2,15E+14	5,98E+09	1,49E+09	0,00E+00
90	0	0	5,56E+12	6,41E+13	1,10E+14	1,04E+14	2,49E+09	0,00E+00	0,00E+00
112,5	0	0	1,33E+12	2,81E+13	5,52E+13	3,46E+13	1,06E+13	0,00E+00	0,00E+00
135	0	0	1,61E+12	4,45E+13	9,02E+13	8,54E+13	2,85E+13	4,98E+12	0,00E+00
157,5	0	0	8,05E+11	2,53E+13	7,68E+13	9,07E+13	4,49E+13	7,65E+11	2,64E+12
180	0	0	4,49E+12	7,88E+13	3,09E+14	5,69E+14	2,54E+10	6,47E+09	9,96E+08
202,5	0	0	2,65E+12	6,84E+13	3,44E+14	7,44E+13	4,08E+13	1,19E+14	2,98E+12
225	0	0	2,55E+12	5,72E+13	2,63E+14	5,30E+14	3,89E+14	1,35E+14	2,33E+13
247,5	0	0	6,58E+12	1,38E+14	6,30E+14	1,01E+15	4,68E+10	1,49E+10	3,49E+09
270	0	0	8,78E+12	1,30E+14	4,90E+14	6,00E+14	2,89E+10	8,47E+09	1,49E+09
292,5	0	0	1,52E+12	3,44E+13	9,86E+13	1,34E+14	8,24E+13	2,66E+13	7,88E+08
315	0	0	1,90E+12	3,85E+13	9,70E+13	1,40E+14	9,05E+12	2,49E+09	4,98E+08
337,5	0	0	7,40E+11	1,73E+12	4,55E+13	6,68E+13	3,62E+12	5,40E+12	0,00E+00

Table 4.5: Energy obtained with each specific wind speed and each specific wind direction.

The total energy per wind direction is the sum of each row and it is expressed in MWh:

E[MWH]	TOTAL
Wind direction [degrees]	
0	4,998E+08
22,5	5,089E+08
45	4,943E+08
67,5	5,292E+08
90	2,834E+08
112,5	1,298E+08
135	2,553E+08
157,5	2,419E+08
180	9,616E+08
202,5	6,521E+08
225	1,400E+09
247,5	1,782E+09
270	1,230E+09
292,5	3,779E+08
315	2,867E+08
337,5	1,237E+08

Table 4.6: Total energy obtained in Fécamp per wind direction.

Considering all the wind directions, the total energy produced during a whole year in the Parc Éolien en mer de Fécamp is the sum of the table above: **9,756E+09 MWh ≈ 1E+10 MWh.**

Taking into account the table 5.3, the following figure is obtained and represents the total energy (in MWh) of Fécamp power plant per wind direction. So, as it can be appreciated, the wind direction that produces more energy during the whole year is **247,5 degrees** (specifically **1,782E+09 MWh**), and the worst one is **337,5 degrees** (specifically **1,237E+08 MWh**).

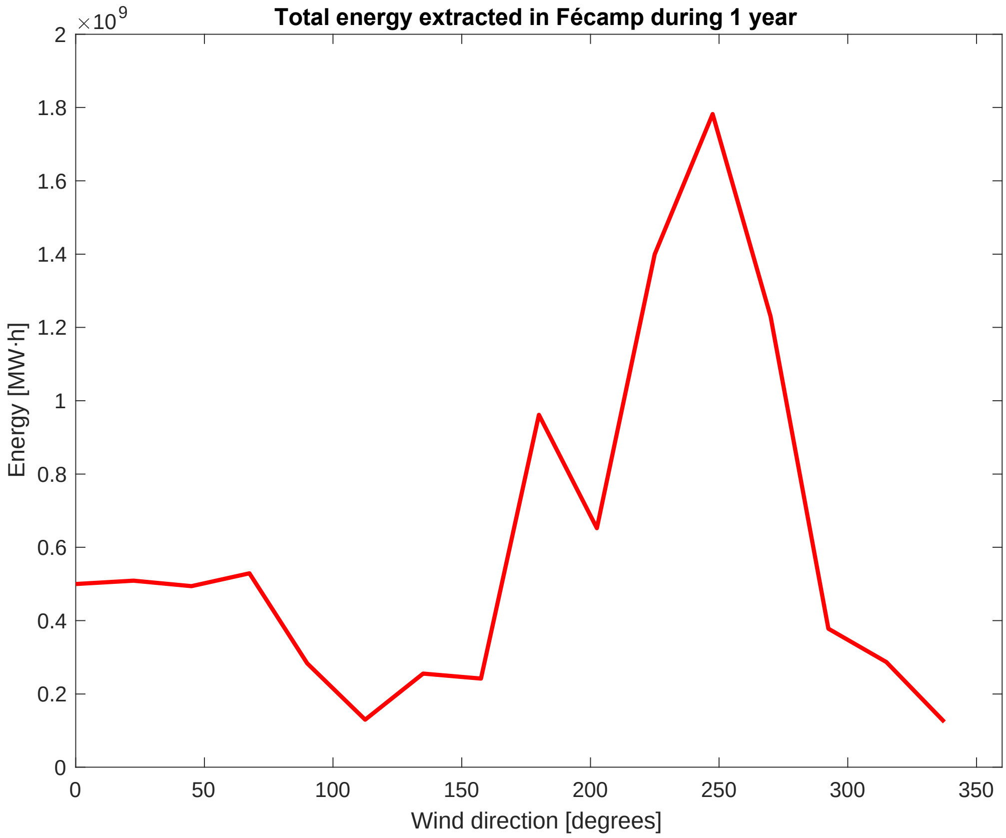


Figure 4.20: Energy obtained per wind direction in Parc Éolien en mer de Fécamp.

Finally, it is interesting to see the influence of the tip speed ratio parameter from a wind turbine. In the figure below, with a free stream wind speed of 9,5 m/s, the power generated in Fécamp for each tip speed ratio (lambda) value from 2 to 9 (with 0,5 increments) is displayed. To see the influence of λ independently of the wind direction, the power shown here is the sum of each power calculated with 12 divisions of the angle alpha from 0 to 360 (with 30 degrees increments).

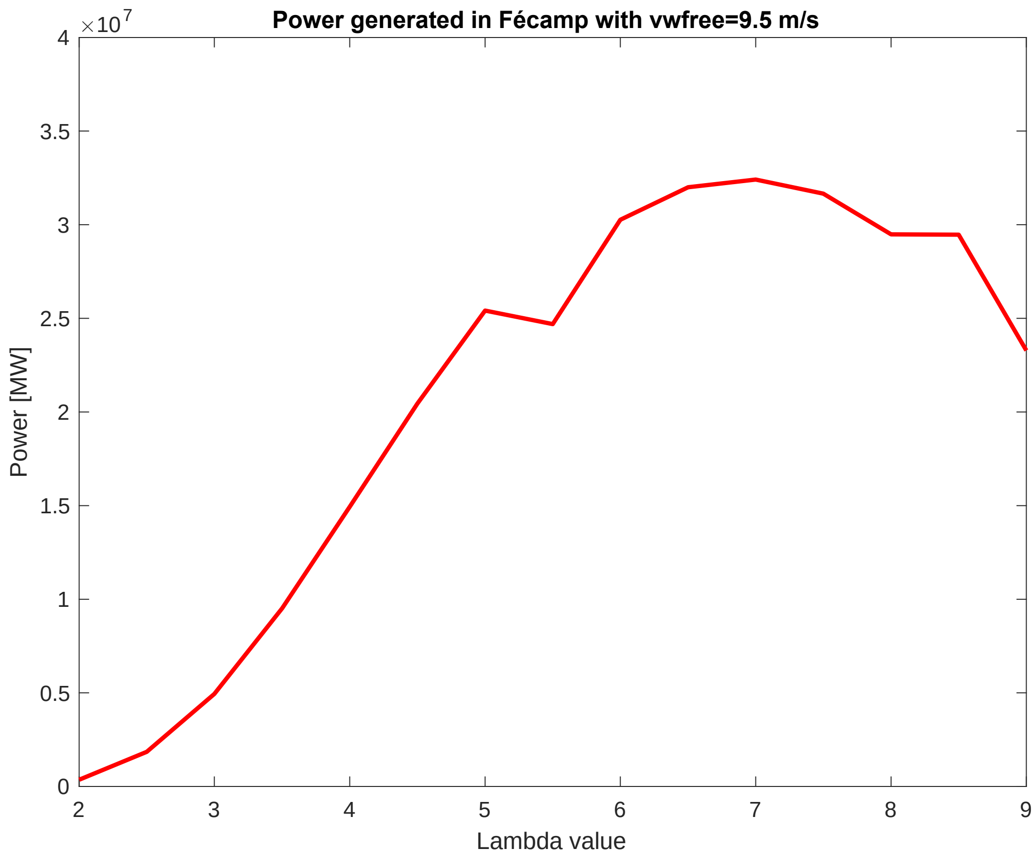


Figure 4.21: Power generated in Fécamp as function of the tip speed ratio value.

As seen, the optimal tip speed ratio is between 6,5 and 7,5. The application case of the Parc Éolien en mer de Fécamp has been analysed with an optimal tip speed ratio of 7,21 because it is the optimal operating point of a single wind turbine which depends on the mechanical characteristics. This figure shows that it is possible to obtain another optimal global operating point of the whole wind power plant by operating some wind turbines at a sub-optimal performance.

To display the effectiveness of each wind turbine separately, the following map of Fécamp is obtained. The energy extracted through a whole year for each wind turbine is calculated making use of the table 2.4, so the quantity of hours per year with a specific wind speed and a specific wind direction is taken into account. As seen, the bigger the size of the bubble is, the more energy is extracted from that wind turbine.

As the difference between the bubbles were not relevant enough to be appreciated in the plot, an exponential function has been applied to all the values in order to exaggerate the differences. In this case, each value has been recalculated as:

New value = 50 ^ (energy per year per wind turbine) (5.1)

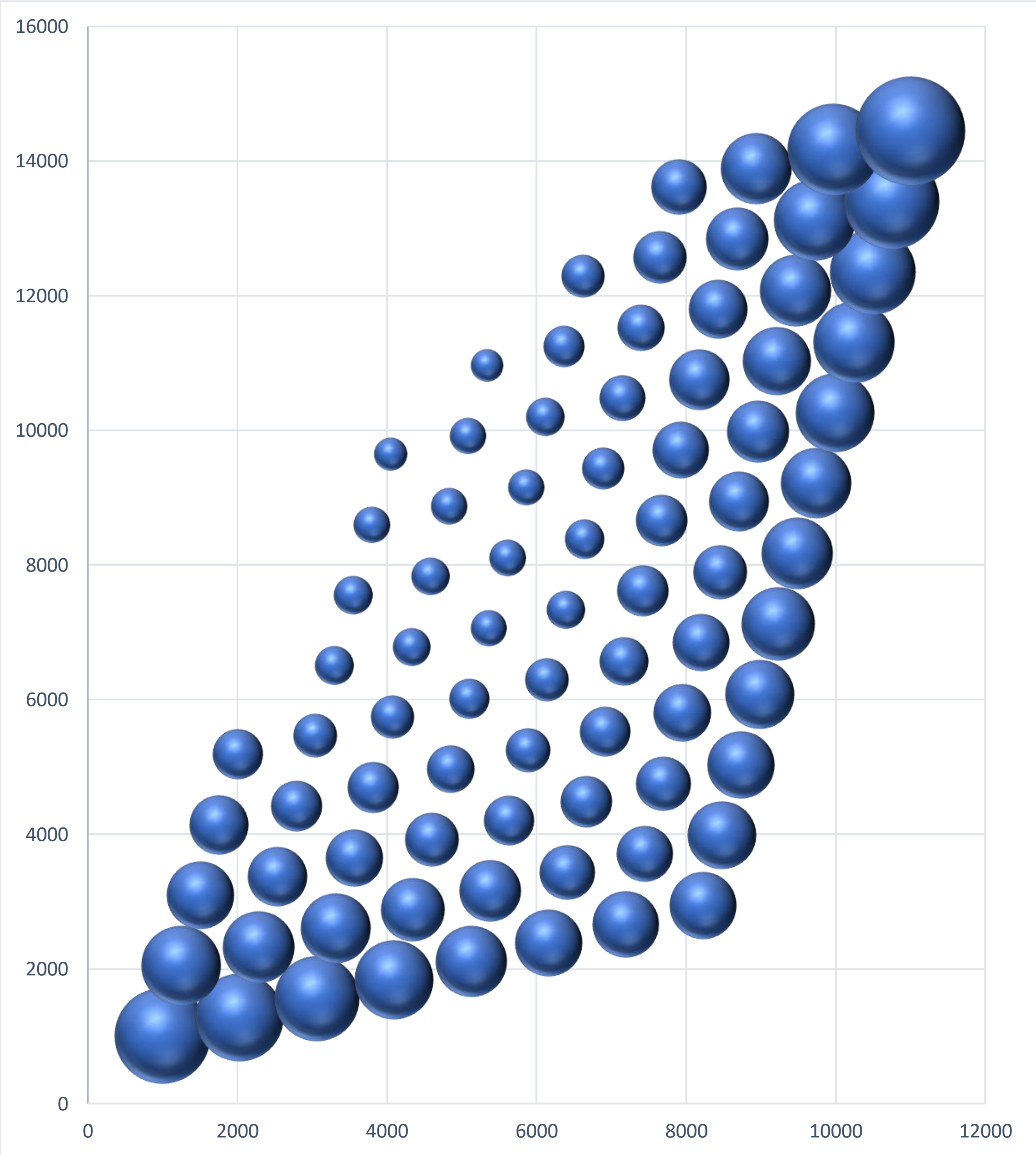


Figure 4.22: Energy generated per wind turbine with real wind data during a year.

Two important things can be noted in this figure. The size of the bubbles is proportional to the energy generated by a wind turbine during a whole year, with real data of wind (both wind directions and wind speeds) and taking into account the wake effect involved in this farm. Thus, the wind turbines at the top and those at the bottom are the ones in better conditions to perform, because the wind blows, mainly, in such directions that make these turbines to be the former ones, so that the wake effect is not present or is lower.

5. Conclusions

In this chapter some conclusions are exposed taking into account the two main cases studied in this project: the optimization case and the realistic application case of Fécamp.

In the sections 4.1.1 and 4.1.2 the optimization methodology has been proved. As said along all this project, this optimization methodology is based on operating each wind turbine in a concrete performance and different from the optimal one, fact that has as a result the increase of the wind speeds that reach each wind turbine and therefore, an increase in the electric generation of the whole wind plant. This methodology has been tested and validated for two simple and ideal wind plant layouts. Another interesting conclusion that results from these simplified models is that each wind direction affects the productivity of the wind plant in a different way. Therefore, the optimal operating point of each wind turbine depends on the angle of the wind direction. This is due to the wake effect produced by the blades of the rest wind turbines.

Afterwards, in the section 4.1.3 a real application case, the Parc Éolien en mer de Fécamp, has been analysed. Due to the complexity of the layout of the plant, this application case has been studied assuming that all the wind turbines operate at their optimal point and not at a concrete operating point, as the conventional operation strategy dictates. It has been studied for all the possible wind directions and with real data of the wind rose in that area (wind rose calculated as the average of the measurement during the last 30 years). Moreover, in this section some results are printed with the help of some functions and plots.

In the figure 4.17 of the section above, it can be appreciated that the wind direction has an important weight on the total power produced by the whole wind plant. Unfortunately, certain wind directions line up a lot of wakes of the wind turbines and this produces a huge decrease of the total power extracted by the whole plant. So, it is concluded that it is important to apply an algorithm that makes each wind turbine operating a different point from the optimal in order to optimize the global energy production, this is the main conclusion validated in this project.

So definitely, as wind turbines are spaced out by a certain distance resulting from a balance between maximising the wind plant energy production, by reducing the wake effects, and minimising the costs associated with the logistics and electrical interconnections between turbines. Figuring out an algorithm that carries out the proposed wind power plant control strategy exposed in this project could be an important way of reducing costs. In principle, this way of reducing the wake effect is much efficient than increasing the spacing between turbines.

BIBLIOGRAPHY

[1] JOSÉ ÁNGEL NARBONA ACEVEDO, "Mejoras En La Implantación De Instalaciones Eólicas En Tierra Firme," p. 127, 2014.

[2] "Parc eolien Fecamp, @ http://parc-eolien-en-mer-de-fecamp.fr." [Online]. Available: http://parc-eolien-en-mer-de-fecamp.fr.

[3] K. Schönleber, S. Rates-Palau, M. De-Prada-Gil, and O. Gomis-Bellmunt, "Reactive power optimization in HVDC-connected wind power plants considering wake effects," *14th Wind Integr. Work.*, 2015.

[4] É. O. des H. Falaises, "Project de Parc Éolien au large de Fécamp," 2013.

[5] "Fécamp weather forecast @ www.meteoblue.com." [Online]. Available: https://www.meteoblue.com/en/weather/forecast/modelclimate/49.892N0.227E0_Europe%2FParis.

[6] C. G. Alías, "Optimización de la generación de un parque eólico considerando efectos de estela," *UB-UPC*, 2013.

[7] A. Junyent-Ferre, "Control of power electronic converters for the operation of wind generation systems under grid disturbances," 2011.

[8] M. De-Prada-Gil, C. G. Alías, O. Gomis-Bellmunt, and A. Sumper, "Maximum wind power plant generation by reducing the wake effect," *Energy Convers. Manag.*, vol. 101, 2015.

[9] "Parc éolien en mer de Fécamp, France @ www.4coffshore.com." [Online]. Available: https://www.4coffshore.com/windfarms/parc-éolien-en-mer-de-fécamp-france-fr34.html.

[10] "Offshore turbine Haliade @ www.ge.com." [Online]. Available: https://www.ge.com/renewableenergy/wind-energy/turbines/offshore-turbine-haliade.

[11] J. Nistal, "Modelado y control dinámico de un Aerogenerador," 2015.

[12] M. Ali, J. Matevosyan, J. V Milanović, and L. Söder, "Effect of wake consideration on estimated cost of wind energy curtailments," *8th Int. Work. Large Scale Integr. Wind Power, Bremen*, no. October, pp. 14–15, 2009.

[13] B. L. R. J. Barthelmie, L. Folkerts, G. C. Larsen, K. Rados, S. C. Pryor, S. T.

Frandsen and G. Schepers., "Comparison of Wake Model Simulations with Offshore Wind Turbine Wake Profiles Measured by Sodar. 2015.," 2005.

[14] N Jensen, "A Note on Wind Generaor Interation," 1983.

[15] M. de Prada Gil, "Design, operation and control of novel electrical concepts for offshore wind power plants," 2014.

ANNEXES

The full code implemented on MATLAB:

1. **First section: Typical Cp vs tip speed ratio curve:**

- **TFG1.m**

```
clc;clear all;

%% Single turbine operation

%Parameters (SI):
rho=1.225; %air
R=63;
A=pi.*(R^2); %swept area
c1=0.5; c2=116; c3=0.4; c4=0; c5=0; c6=5; c7=21; c8=0.08; c9=0.035;
%constants for equation Cp *
wt=2.2588;
vw=9.5; %average wind speed at hub height [9-10m/s] *
teta_pitch=0; %blade pitch angle

lambda=(R.*wt)./vw %tip speed ratio
Lambda=((lambda+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.^3
)-c9*(lambda+c8.*teta_pitch));
Cp=c1.*(c2.*(1./Lambda)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-c6).*exp(-
c7.*(1./Lambda))

%Optimal point
Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

%% Typical Cp vs tip speed ratio curve

%Parameters (SI):
rho=1.225; %air
R=75;
A=pi.*(R.^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003; %constants for equation Cp *
wt=2.2588;
vw=9.5; %average wind speed at hub height [9-10m/s] *
teta_pitch=0;

lambda=0:0.1:14; %tip speed ratio
Lambda=((lambda+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.^3
)-c9*(lambda+c8.*teta_pitch));
Cp=c1.*(c2.*(1./Lambda)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-c6).*exp(-
c7.*(1./Lambda));
Cp_positiva=subplus(Cp);
plot(lambda,Cp_positiva,'LineWidth',3);
xlabel('lambda');
ylabel('Cp');
grid on
axis([0 14 0 0.5]);
```

```
figure(1)

Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

Cp_opt2=((c1*c2)/c7)*exp(-(c2+(c6*c7))/c2);
lambda_opt2=1/(c9+(c6/c2)+(1/c7));

%% CT vs lambda

Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
lambda=2:0.1:9;
plot(lambda,Ct,'LineWidth',3);
xlabel('lambda');
ylabel('Ct');
grid on
axis([0 14 0 1]);
figure(1)
```

2. Second section: 3 WT case

- **TFG2.m**

```
clc; clear all;

%% 3 WT case (7D separation)

%STEP 1:

%Parameters WT1 (SI):
rho=1.225; %air
R=63;
A=pi.*(R^2); %swept area
c1_1=0.73; c2_1=151; c3_1=0.58; c4_1=0.002; c5_1=2.14; c6_1=13.2;
c7_1=18.4; c8_1=-0.02; c9_1=-0.003; %constants for equation Cp *
wt1=2.2588;
vw1=9.5;
```

```
teta_pitch1=0;

lambda1=2:0.1:9; %tip speed ratio WT1
Lambda1=((lambda1+c8_1.*teta_pitch1).*(1+teta_pitch1.^3))./((1+teta_pi
tch1.^3)-c9_1*(lambda1+c8_1.*teta_pitch1));
Cp1=c1_1.*(c2_1.*(1./Lambda1)-c3_1.*teta_pitch1-
c4_1.*(teta_pitch1.^c5_1)-c6_1).*exp(-c7_1.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));

[lambda1,lambda2]=meshgrid(2:0.1:9,2:0.1:9); %tip speed ratio WT2
surf(lambda2,lambda1,P_WT1)
xlim([2 10]);
ylim([2 10]);
%zlim([0 3]);
colormap jet
shading interp
colorbar
xlabel('lambda_2'); ylabel('lambda_1'); zlabel('Power WT1');
figure()

%% STEP 2:

%Parameters WT2 (SI):
rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1_2=0.73; c2_2=151; c3_2=0.58; c4_2=0.002; c5_2=2.14; c6_2=13.2;
c7_2=18.4; c8_2=-0.02; c9_2=-0.003; %constants for equation Cp !!
wt2=2.2588;
vw1=9.5;
teta_pitch2=0;
lambda2=2:0.1:9; %tip speed ratio WT1
Lambda2=((lambda2+c8_2.*teta_pitch2).*(1+teta_pitch2.^3))./((1+teta_pi
tch2.^3)-c9_2*(lambda2+c8_2.*teta_pitch2));
Cp2=c1_2.*(c2_2.*(1./Lambda2)-c3_2.*teta_pitch2-
c4_2.*(teta_pitch2.^c5_2)-c6_2).*exp(-c7_2.*(1./Lambda2));

Cp2_2=[0.0191304033218519,0.0212435209434604,0.0235276775506929,0.0259
931128708419,0.0286501859762797,0.0315093611541758,0.0345811963540072,
0.0378763337258213,0.0414054918644017,0.0451794594542802,0.04920909007
30428,0.0535052979595786,
0.0580790545928385,0.0629413859576538,0.0681033703990041,0.07368724281
03668,0.0797130675162090,0.0862232933078461,0.0932852014663414,0.10099
3596788332,0.109402073175567,0.118543041489955,0.128386880013709,0.138
876629816002,0.149926711566370,0.161458284957121,0.173380945343715,0.1
85614069609024,0.198075677637596,0.210691261846497,0.223397647138997,0
.236125482587739,0.248816358652213,0.261421230388339,0.273883753880445
,0.286165786794813,0.298219258605052,0.310013106871736,0.3215082646702
79,0.332674678181208,0.343483051314469,0.353906935901936,0.36392531396
7969,0.373514282731785,0.382644493827965,0.391316933150126,0.399475346
959082,0.407046752892542,0.413952314871398,0.420700633575729,0.4269286
47089748,0.432691954468416,0.438017924803260,0.442988166779286,0.44755
0736300269,0.451849483283848,0.455892833866409,0.459536600202486,0.462
766151010501,0.465611524457378,0.467973205911285,0.470035210393983,0.4
71581137460235,0.472630644964611,0.473336081086407,0.473408925639723,0
.472959469377894,0.472169106140777,0.470606150792860,0.468437064408651
,0.465628036767700];
```

```
Ct1=[0.121803150380010,0.125991426879747,0.130372866788154,0.134947794
202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1551893
01827121,0.160736906779912,0.166480116493183,0.172419283098105,0.17855
4753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.205
122378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.2
35905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0
.272757594242936,0.283007408104841,0.293646835724271,0.304653069492279
,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874525394
19,0.364556380282990,0.377383832357742,0.390456906615975,0.40375289455
5614,0.417266029429188,0.430970459886915,0.444858279381295,0.458909231
955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5164274
01248424,0.531048382940466,0.545703059988886,0.560416750539903,0.57506
3029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.630
471356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.6
78127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0
.722307736139963,0.733079282358001,0.743634250978271,0.754095088731837
,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657004832
69,0.802933289131714,0.812123022351999,0.820863428582445,0.82931445573
6052,0.837463579391426;];

plot(lambda2,Cp2)
xlabel('lambda2');ylabel('Cp2');
figure()
plot(lambda2,Cp2_2)
xlabel('lambda2');ylabel('Cp2_2');
figure()
plot(lambda2,Ct1)
xlabel('lambda2');ylabel('Ct1');
figure()

k=0.04; %Jensen for offshore case
x=7.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct1)));

Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2_tras*(vw2.^3);

[lambda1,lambda2]=meshgrid(2:0.1:9,2:0.1:9); %tip speed ratio WT2
surf(lambda2,lambda1,P_WT2./(10^6));
xlim([2 10]);
ylim([2 10]);
%zlim([0 3]);
xlabel('lambda_2'); ylabel('lambda_1'); zlabel('Power WT2 (MW)');
colormap jet
shading interp
colorbar
figure()

%% STEP 3:

%Parameters WT3 (SI):
rho=1.225; %air
R=63; %D/2 minus hub (blades length)
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003; %constants for equation Cp *
teta_pitch3=0;
vw1=9.5;
```

```
%Optimal point WT3
Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

lambda1=2:0.1:9;
lambda2=2:0.1:9;

Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];

k=0.04; %Jensen for offshore case
x=7.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct)));
vw2_tras=vw2';
vw3_single=vw2'.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct)));

vw3_multiple=vw1.*(1-sqrt(((1-(vw2./vw1)).^2)+(1-
(vw3_single./vw1).^2)));
P_WT3=0.5.*rho.*A.*Cp3_opt*(vw3_single.^3);

[lambda1,lambda2]=meshgrid(2:0.1:9,2:0.1:9); %tip speed ratio WT2
surfc(lambda2,lambda1,P_WT3./(10^6));
xlim([2 10]);
ylim([2 10]);
%zlim([0 3]);
xlabel('lambda_2'); ylabel('lambda_1'); zlabel('Power WT3 (MW)');
colormap jet
shading interp
colorbar
figure()

%% STEP 4

P_TOT=P_WT1+P_WT2+P_WT3;
[lambda1,lambda2]=meshgrid(2:0.1:9,2:0.1:9); %tip speed ratio WT2
surfc(lambda2,lambda1,P_TOT./(10^6));
xlim([2 10]);
ylim([2 10]);
%zlim([0 3]);
xlabel('lambda_2'); ylabel('lambda_1'); zlabel('Power TOTAL (MW)');
colormap jet
shading interp
colorbar
```

```
[M,I]=max(P_TOT(:));
[I_row, I_col] = ind2sub(size(P_TOT),I);

%Resultados:
lambda3_final=lambda3_opt
lambda2_final=lambda2(I_row,I_col)
lambda1_final=lambda1(I_row,I_col)
Cp3_final=Cp3_opt
Lambda2_final=((lambda2_final+c8_2.*teta_pitch2).*(1+teta_pitch2.^3)).
/((1+teta_pitch2.^3)-c9_2*(lambda2_final+c8_2.*teta_pitch2));
Cp2_final=c1_2.*(c2_2.*(1./Lambda2_final)-c3_2.*teta_pitch2-
c4_2.*(teta_pitch2.^c5_2)-c6_2).*exp(-c7_2.*(1./Lambda2_final))
Lambda1_final=((lambda1_final+c8_1.*teta_pitch1).*(1+teta_pitch1.^3)).
/((1+teta_pitch1.^3)-c9_1*(lambda1_final+c8_1.*teta_pitch1));
Cp1_final=c1_1.*(c2_1.*(1./Lambda1_final)-c3_1.*teta_pitch1-
c4_1.*(teta_pitch1.^c5_1)-c6_1).*exp(-c7_1.*(1./Lambda1_final))

v1_final=(2*(10^6))./(0.5.*rho.*A.*Cp1_final)
v2_final=(2*(10^6))./(0.5.*rho.*A.*Cp2_final)
v3_final=(2*(10^6))./(0.5.*rho.*A.*Cp3_final)

omega1_final=(lambda1_final.*v1_final)./R
omega2_final=(lambda2_final.*v1_final)./R
omega3_final=(lambda3_final.*v1_final)./R
```

3. Third section: 9 WT case

- **tfg3_a.m**

```
function casa = tfg3_a(vw1)

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle
lambda=2:0.01:9;

% - Upwind turbines (3 WT)

lambda1=2:0.01:9; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));


% - WTs affected by single wake (3 WT)

lambda2=2:0.01:9;
```

```
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));


Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case
x=7.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt)));
Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2_tras*(vw2.^3);

% - WTs affected by multiple wake (3 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

vw3_single=vw2'.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt)));

P_WT3=0.5.*rho.*A.*Cp3_opt*(vw3_single.^3);

% - Resultados

P_TOT_casA_no=3.*P_WT1+3.*P_WT2+3.*P_WT3;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casA=3.*max(P_WT1(:))+3.*max(P_WT2(:))+3.*max(P_WT3(:));
```

ETSEIB

```
[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M,I]=max(P_TOT_casA(:));
[I_row, I_col] = ind2sub(size(P_TOT_casA),I);

lambda3_final_casA=lambda3_opt;
lambda2_final_casA=lambda2(I_row,I_col);
lambda1_final_casA=lambda1(I_row,I_col);
Cp3_final_casA=Cp3_opt;
Lambda2_final_casA=((lambda2_final_casA+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casA+c8.*teta_pitch));
Cp2_final_casA=c1.*(c2.*(1./Lambda2_final_casA)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casA));
Lambda1_final_casA=((lambda1_final_casA+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casA+c8.*teta_pitch));
Cp1_final_casA=c1.*(c2.*(1./Lambda1_final_casA)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casA));

casa=M;

end
```

- **tfg3_a2.m**

```
function casa2 = tfg3_a2(vw1)

rho=1.225; %air
R=63; %D/2 minus hub (blades length) *TMdPG1de1.pdf
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0;
lambda=2:0.01:9;
Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

% - Upwind turbines (3 WT)

lambda1=lambda_opt; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));
%igual q Cp_opt
P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));


% - WTs affected by single wake (3 WT)

lambda2=lambda_opt;
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));
```

```
lambda_optima_1decimal=round(lambda_opt,1);
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));

k=0.04; %Jensen for offshore case
x=7.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct_lambda_optima)));

%Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2*(vw2.^3);

% - WTs affected by multiple wake (3 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

vw3_single=vw2.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-
Ct_lambda_optima))); %average wind speed at hub height of WT2 with
wake effect

P_WT3=0.5.*rho.*A.*Cp3_opt*(vw3_single.^3);

% - Resultados

P_TOT_casA_no=3.*P_WT1+3.*P_WT2+3.*P_WT3;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casA=3.*max(P_WT1(:))+3.*max(P_WT2(:))+3.*max(P_WT3(:));
```

```
[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M,I]=max(P_TOT_casA(:));
[I_row, I_col] = ind2sub(size(P_TOT_casA),I);

lambda3_final_casA=lambda3_opt;
lambda2_final_casA=lambda2(I_row,I_col);
lambda1_final_casA=lambda1(I_row,I_col);
Cp3_final_casA=Cp3_opt;
Lambda2_final_casA=((lambda2_final_casA+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casA+c8.*teta_pitch));
Cp2_final_casA=c1.*(c2.*(1./Lambda2_final_casA)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casA));
Lambda1_final_casA=((lambda1_final_casA+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casA+c8.*teta_pitch));
Cp1_final_casA=c1.*(c2.*(1./Lambda1_final_casA)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casA));

casa2=M;

end
```

- **tfg3_b.m**

```
function casb = tfg3_b(vw1)

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle !!
lambda=2:0.01:9;

% - Upwind turbines (3 WT)

lambda1=2:0.01:9;
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));

% - WTs affected by single wake (3 WT)

lambda2=2:0.01:9;
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));


Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
```

```
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case
x=5.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt)));

Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2_tras*(vw2.^3);

% - WTs affected by multiple wake (3 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

vw3_single=vw2'.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt)));

P_WT3=0.5.*rho.*A.*Cp3_opt*(vw3_single.^3);

% - Resultados

P_TOT_casB_no=3.*P_WT1+3.*P_WT2+3.*P_WT3;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casB=3.*max(P_WT1(:))+3.*max(P_WT2(:))+3.*max(P_WT3(:));

[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M_casB,I]=max(P_TOT_casB(:));
[I_row, I_col] = ind2sub(size(P_TOT_casB),I);

lambda3_final_casB=lambda3_opt;
lambda2_final_casB=lambda2(I_row,I_col);
```

```
lambda1_final_casB=lambda1(I_row,I_col);
Cp3_final_casB=Cp3_opt;
Lambda2_final_casB=((lambda2_final_casB+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casB+c8.*teta_pitch));
Cp2_final_casB=c1.*(c2.*(1./Lambda2_final_casB)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casB));
Lambda1_final_casB=((lambda1_final_casB+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casB+c8.*teta_pitch));
Cp1_final_casB=c1.*(c2.*(1./Lambda1_final_casB)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casB));

casb=M_casB;

end
```

- **tfg3_b2.m**

```
function casb2 = tfg3_b2(vw1)

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle
lambda=2:0.01:9;
Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

% - Upwind turbines (3 WT)

lambda1=lambda_opt; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));

% - WTs affected by single wake (3 WT)

lambda2=lambda_opt;
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));

lambda_optima_1decimal=round(lambda_opt,1);
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
```

```
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));

k=0.04; %Jensen for offshore case
x=5.*D; %distance between WT
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake

%Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2*(vw2.^3);

% - WTs affected by multiple wake (3 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

vw3_single=vw2.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-
Ct_lambda_optima))); %average wind speed at hub height of WT2 with
wake effect

P_WT3=0.5.*rho.*A.*Cp3_opt*(vw3_single.^3);

% - Resultados

P_TOT_casB_no=3.*P_WT1+3.*P_WT2+3.*P_WT3;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casB=3.*max(P_WT1(:))+3.*max(P_WT2(:))+3.*max(P_WT3(:));

[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M_casB,I]=max(P_TOT_casB(:));
[I_row, I_col] = ind2sub(size(P_TOT_casB),I);

lambda3_final_casB=lambda3_opt;
```

```
lambda2_final_casB=lambda2(I_row,I_col);
lambda1_final_casB=lambda1(I_row,I_col);
Cp3_final_casB=Cp3_opt;
Lambda2_final_casB=((lambda2_final_casB+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casB+c8.*teta_pitch));
Cp2_final_casB=c1.*(c2.*(1./Lambda2_final_casB)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casB));
Lambda1_final_casB=((lambda1_final_casB+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casB+c8.*teta_pitch));
Cp1_final_casB=c1.*(c2.*(1./Lambda1_final_casB)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casB));

casb2=M_casB;

end
```

- **tfg3_c.m**

```
function casc = tfg3_c(vw1)

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0;
lambda=2:0.01:9;

% - Upwind turbines (5 WT)

lambda1=2:0.01:9; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=0.5.*rho.*A.*Cp1.*(vw1.^3);

% - WTs affected by partial wake (3 WT)

lambda2=2:0.01:9;
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));

Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
```

```
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case

alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd=5.*D; %x distance between WT **VARIABLE**
yd=7.*D; %y distance between WT **VARIABLE**

xa=((yd.^2)./(yd+(cos(beta).*R)));
ya=(xd.*sqrt(((yd./cos(alpha)).^2)-
(yd.^2)))./(sqrt(((yd./cos(alpha)).^2)-(yd.^2))-R.*sin(beta));
x=sqrt((xa.^2)+(ya.^2)); %distance between free-stream-WT and point
where the single wake affects (to the next WT)
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt))); %average wind
speed at hub height of WT2 with wake effect, Jensen single wake

Dx=D+2.*k.*x;
r2=R;
r1=Dx./2;
d=sqrt((xa-7.*D).^(2)+(ya-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
betatjtk=Ashad./Awind;
vtj=vw1.*(1-sqrt(betatjtk.*(1-vw2./vw1).^2));

Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2.*(vtj.^3);


% - WTs affected by multiple wake (1 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

%partial wake de WT1 sobre WT9: (subindex 19):
Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
```

```
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case
alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd19=2.*5.*D; %x distance between WT **VARIABLE**
yd19=2.*7.*D; %y distance between WT **VARIABLE**
xa19=((yd19.^2)./(yd19+(cos(beta).*R)));
ya19=(xd19.*sqrt(((yd19./cos(alpha)).^2)-
(yd19.^2)))./(sqrt(((yd19./cos(alpha)).^2)-(yd19.^2))-R.*sin(beta));
x19=sqrt((xa19.^2)+(ya19.^2));
vw219=vw1.*(1-((D./(D+2.*k.*x19)).^2).*(1-sqrt(1-vCt))); %average wind
speed at hub height of WT2 with wake effect, Jensen single wake
Dx19=D+2.*k.*x19;
r2=R;
r119=Dx19./2;
d19=sqrt((xa19-7.*D).^(2)+(ya19-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad19=(acos(((r119.^2)+(d19.^2)-
(r2.^2))./(2.*r119.*d19)).*(r119.^2))+(acos(((r2.^2)+(d19.^2)-
(r119.^2))./(2.*r2.*d19)).*(r2.^2))-(sin(acos(((r119.^2)+(d19.^2)-
(r2.^2))./(2.*r119.*d19)))).*r119.*d19;
betatjtk19=Ashad19./Awind;
vtj19=vw1.*(1-sqrt(betatjtk19.*(1-vw219./vw1).^2));

Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case
alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd59=5.*D; %x distance between WT **VARIABLE**
yd59=7.*D; %y distance between WT **VARIABLE**
xa59=((yd59.^2)./(yd59+(cos(beta).*R)));
ya59=(xd59.*sqrt(((yd59./cos(alpha)).^2)-
```

```
(yd59.^2))./(sqrt(((yd59./cos(alpha)).^2)-(yd59.^2))-R.*sin(beta));
x59=sqrt((xa59.^2)+(ya59.^2)); %distance between free-stream-WT and
point where the single wake affects (to the next WT)
vw259=vtj.*(1-((D./(D+2.*k.*x59)).^2).*(1-sqrt(1-vCt))); %average wind
speed at hub height of WT2 with wake effect, Jensen single wake
Dx59=D+2.*k.*x59;
r2=R;
r159=Dx59./2;
d59=sqrt((xa59-7.*D).^(2)+(ya59-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad59=(acos(((r159.^2)+(d59.^2)-
(r2.^2))./(2.*r159.*d59)).*(r159.^2))+(acos(((r2.^2)+(d59.^2)-
(r159.^2))./(2.*r2.*d59)).*(r2.^2))-(sin(acos(((r159.^2)+(d59.^2)-
(r2.^2))./(2.*r159.*d59)))).*r159.*d59;
betatjtk59=Ashad59./Awind;
vtj59=vtj.*(1-sqrt(betatjtk59.*(1-vw259./vtj).^2));
vwt9=vw1.*(1-sqrt(((1-vtj19./vw1).^2)+((1-vtj59./vw1).^2)));
P_WT3=0.5.*rho.*A.*Cp3_opt.*(vwt9.^3);

% - Resultados

P_TOT_casC_no=5.*P_WT1+3.*P_WT2+1.*P_WT3;


if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casC=5.*max(P_WT1(:))+3.*max(P_WT2(:))+1.*max(P_WT3(:));

%[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9)); %probem

[M_casC,I]=max(P_TOT_casC(:));
[I_row, I_col] = ind2sub(size(P_TOT_casC),I);

lambda3_final_casC=lambda3_opt;
lambda2_final_casC=lambda2(I_row,I_col);
lambda1_final_casC=lambda1(I_row,I_col);
Cp3_final_casC=Cp3_opt;
Lambda2_final_casC=((lambda2_final_casC+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casC+c8.*teta_pitch));
Cp2_final_casC=c1.*(c2.*(1./Lambda2_final_casC)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casC));
Lambda1_final_casC=((lambda1_final_casC+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casC+c8.*teta_pitch));
Cp1_final_casC=c1.*(c2.*(1./Lambda1_final_casC)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casC));


casc = M_casC;
```

ETSEIB

```
end
```

- **tfg3_c2.m**

```
function casc2 = tfg3_c2(vw1)

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle !!
lambda=2:0.01:9;
Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

% - Upwind turbines (5 WT)

lambda1=lambda_opt; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=0.5.*rho.*A.*Cp1.*(vw1.^3);

% - WTs affected by partial wake (3 WT)

lambda2=lambda_opt;
Lambda2=((lambda2+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda2+c8.*teta_pitch));
Cp2=c1.*(c2.*(1./Lambda2)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda2));

lambda_optima_1decimal=round(lambda_opt,1); %=7.2 cas TFG4
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));

k=0.04; %Jensen for offshore case
```

```
alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd=5.*D; %x distance between WT **VARIABLE**
yd=7.*D; %y distance between WT **VARIABLE**

xa=((yd.^2)./(yd+(cos(beta).*R))); %en gen√©rico, A=(xa,ya) punt
intersecci√≥ vector vent amb vector turbina
ya=(xd.*sqrt(((yd./cos(alpha)).^2)-
(yd.^2)))./(sqrt(((yd./cos(alpha)).^2)-(yd.^2))-R.*sin(beta));
x=sqrt((xa.^2)+(ya.^2)); %distance between free-stream-WT and point
where the single wake affects (to the next WT)
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake

Dx=D+2.*k.*x;
r2=R;
r1=Dx./2;
d=sqrt((xa-7.*D).^(2)+(ya-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
betatjtk=Ashad./Awind;
vtj=vw1.*(1-sqrt(betatjtk.*(1-vw2./vw1).^2));

%Cp2_tras=Cp2';
P_WT2=0.5.*rho.*A.*Cp2.*(vtj.^3);


% - WTs affected by multiple wake (1 WT)

Cp3_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda3_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

k=0.04; %Jensen for offshore case
alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd19=2.*5.*D; %x distance between WT **VARIABLE**
yd19=2.*7.*D; %y distance between WT **VARIABLE**
xa19=((yd19.^2)./(yd19+(cos(beta).*R)));
ya19=(xd19.*sqrt(((yd19./cos(alpha)).^2)-
(yd19.^2)))./(sqrt(((yd19./cos(alpha)).^2)-(yd19.^2))-R.*sin(beta));
x19=sqrt((xa19.^2)+(ya19.^2)); %distance between free-stream-WT and
point where the single wake affects (to the next WT)
vw219=vw1.*(1-((D./(D+2.*k.*x19)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake
Dx19=D+2.*k.*x19;
r2=R;
r119=Dx19./2;
d19=sqrt((xa19-7.*D).^(2)+(ya19-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad19=(acos(((r119.^2)+(d19.^2)-
(r2.^2))./(2.*r119.*d19)).*(r119.^2))+(acos(((r2.^2)+(d19.^2)-
(r119.^2))./(2.*r2.*d19)).*(r2.^2))-(sin(acos(((r119.^2)+(d19.^2)-
(r2.^2))./(2.*r119.*d19)))).*r119.*d19;
betatjtk19=Ashad19./Awind;
vtj19=vw1.*(1-sqrt(betatjtk19.*(1-vw219./vw1).^2));
```

```
k=0.04; %Jensen for offshore case
alpha=30.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd59=5.*D; %x distance between WT **VARIABLE**
yd59=7.*D; %y distance between WT **VARIABLE**
xa59=((yd59.^2)./(yd59+(cos(beta).*R)));
ya59=(xd59.*sqrt(((yd59./cos(alpha)).^2)-
(yd59.^2)))./(sqrt(((yd59./cos(alpha)).^2)-(yd59.^2))-R.*sin(beta));
x59=sqrt((xa59.^2)+(ya59.^2)); %distance between free-stream-WT and
point where the single wake affects (to the next WT)
vw259=vtj.*(1-((D./(D+2.*k.*x59)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake
Dx59=D+2.*k.*x59;
r2=R;
r159=Dx59./2;
d59=sqrt((xa59-7.*D).^(2)+(ya59-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad59=(acos(((r159.^2)+(d59.^2)-
(r2.^2))./(2.*r159.*d59)).*(r159.^2))+(acos(((r2.^2)+(d59.^2)-
(r159.^2))./(2.*r2.*d59)).*(r2.^2))-(sin(acos(((r159.^2)+(d59.^2)-
(r2.^2))./(2.*r159.*d59)))).*r159.*d59;
betatjtk59=Ashad59./Awind;
vtj59=vtj.*(1-sqrt(betatjtk59.*(1-vw259./vtj).^2));

vwt9=vw1.*(1-sqrt(((1-vtj19./vw1).^2)+((1-vtj59./vw1).^2)));
P_WT3=0.5.*rho.*A.*Cp3_opt.*(vwt9.^3);

% - Resultados

P_TOT_casC_no=5.*P_WT1+3.*P_WT2+1.*P_WT3;


if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

if max(P_WT3(:))>5000000
    P_WT3=5000000;
end

P_TOT_casC=5.*max(P_WT1(:))+3.*max(P_WT2(:))+1.*max(P_WT3(:));


[M_casC,I]=max(P_TOT_casC(:));
[I_row, I_col] = ind2sub(size(P_TOT_casC),I);

lambda3_final_casC=lambda3_opt;
lambda2_final_casC=lambda2(I_row,I_col);
lambda1_final_casC=lambda1(I_row,I_col);
Cp3_final_casC=Cp3_opt;
Lambda2_final_casC=((lambda2_final_casC+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casC+c8.*teta_pitch));
Cp2_final_casC=c1.*(c2.*(1./Lambda2_final_casC)-c3.*teta_pitch-
```

```
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casC));
Lambda1_final_casC=((lambda1_final_casC+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casC+c8.*teta_pitch));
Cp1_final_casC=c1.*(c2.*(1./Lambda1_final_casC)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casC));


casc2 = M_casC;

end
```

- **tfg3_d.m**

```
function casd = tfg3_d(vw1)

rho=1.225; %air
R=63; %D/2 minus hub (blades length) *TMdPG1de1.pdf
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle !!


lambda=2:0.01:9;

% - Upwind turbines (7 WT)

lambda1=2:0.01:9; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));

% - WTs affected by partial wake (2 WT)

Cp2_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda2_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);


Ct=[0.121803150380010,0.125991426879747,0.130372866788154,0.1349477942
02068,0.139716548938385,0.144679480924098,0.149836945852410,0.15518930
1827121,0.160736906779912,0.166480116493183,0.172419283098105,0.178554
753945989,0.184886870773109,0.191415969096183,0.198142377789023,0.2051
22378008047,0.212363973950428,0.219884823744599,0.227716574147805,0.23
5905752851484,0.244486035989457,0.253486104334996,0.262911557943879,0.
272757594242936,0.283007408104841,0.293646835724271,0.304653069492279,
0.316010028258569,0.327696636021499,0.339693939413586,0.35198745253941
9,0.364556380282990,0.377383832357742,0.390456906615975,0.403752894555
614,0.417266029429188,0.430970459886915,0.444858279381295,0.4589092319
55795,0.473108898876679,0.487439791375504,0.501882662099191,0.51642740
1248424,0.531048382940466,0.545703059988886,0.560416750539903,0.575063
029023819,0.589679761955707,0.602954103820260,0.617061697842447,0.6304
71356340617,0.643136907473184,0.654967413059928,0.666835976216401,0.67
8127641069006,0.689303051544152,0.700396551059694,0.711407227335882,0.
```

```
722307736139963,0.733079282358001,0.743634250978271,0.754095088731837,
0.764282533105000,0.774264611585461,0.784142619928012,0.79366570048326
9,0.802933289131714,0.812123022351999,0.820863428582445,0.829314455736
052,0.837463579391426;];
xCt=1:0.1:71;
vCt=interp1(Ct,xCt);
k=0.04; %Jensen for offshore case

alpha=60.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd=2.*5.*D; %x distance between WT **VARIABLE**
yd=7.*D; %y distance between WT **VARIABLE**

xa=((yd.^2)./(yd+(cos(beta).*R)));
ya=(xd.*sqrt(((yd./cos(alpha)).^2)-
(yd.^2)))./(sqrt(((yd./cos(alpha)).^2)-(yd.^2))-R.*sin(beta));
x=sqrt((xa.^2)+(ya.^2));
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-vCt))); %average wind
speed at hub height of WT2 with wake effect, Jensen single wake

Dx=D+2.*k.*x;
r2=R;
r1=Dx./2;
d=sqrt((xa-7.*D).^(2)+(ya-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
betatjtk=Ashad./Awind;
vtj=vw1.*(1-sqrt(betatjtk.*(1-vw2./vw1).^2));

P_WT2=meshgrid(0.5.*rho.*A.*Cp2_opt*(vtj.^3));

% - WTs affected by multiple wake (0 WT)

% - Resultados

P_TOT_casD_no=7.*P_WT1+2.*P_WT2;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

P_TOT_casD=7.*max(P_WT1(:))+2.*max(P_WT2(:));

[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M_casD,I]=max(P_TOT_casD(:));
[I_row, I_col] = ind2sub(size(P_TOT_casD),I);


lambda2_final_casD=lambda2_opt;
lambda1_final_casD=lambda1(I_row,I_col);
Lambda2_final_casD=((lambda2_final_casD+c8.*teta_pitch).*(1+teta_pitch
```

```
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casD+c8.*teta_pitch));
Cp2_final_casD=c1.*(c2.*(1./Lambda2_final_casD)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casD));
Lambda1_final_casD=((lambda1_final_casD+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casD+c8.*teta_pitch));
Cp1_final_casD=c1.*(c2.*(1./Lambda1_final_casD)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casD));

casd=M_casD;

end
```

- **tfg3_d2.m**

```
function casd2 = tfg3_d2(vw1)

rho=1.225; %air
R=63; %D/2 minus hub (blades length) *TMdPG1de1.pdf
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle !! (cuando no estamos en Pnom, es 0?)
lambda=2:0.01:9;
Cp_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

% - Upwind turbines (7 WT)

lambda1=lambda_opt; %tip speed ratio WT1
Lambda1=((lambda1+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_pitch.
^3)-c9*(lambda1+c8.*teta_pitch));
Cp1=c1.*(c2.*(1./Lambda1)-c3.*teta_pitch-c4.*(teta_pitch.^c5)-
c6).*exp(-c7.*(1./Lambda1));

P_WT1=meshgrid(0.5.*rho.*A.*Cp1.*(vw1.^3));

% - WTs affected by partial wake (2 WT)

Cp2_opt=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
lambda2_opt=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);

lambda_optima_1decimal=round(lambda_opt,1); %=7.2 cas TFG4
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
```

```
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));

k=0.04; %Jensen for offshore case

alpha=60.*(pi./180); %radians **VARIABLE**
beta=(pi./2)-alpha; %radians
xd=2.*5.*D; %x distance between WT **VARIABLE**
yd=7.*D; %y distance between WT **VARIABLE**

xa=((yd.^2)./(yd+(cos(beta).*R)));
ya=(xd.*sqrt(((yd./cos(alpha)).^2)-
(yd.^2)))./(sqrt(((yd./cos(alpha)).^2)-(yd.^2))-R.*sin(beta));
x=sqrt((xa.^2)+(ya.^2)); %distance between free-stream-WT and point
where the single wake affects (to the next WT)
vw2=vw1.*(1-((D./(D+2.*k.*x)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake

Dx=D+2.*k.*x;
r2=R;
r1=Dx./2;
d=sqrt((xa-7.*D).^(2)+(ya-10.*D).^(2));
Awind=pi.*(r2.^2);
Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
betatjtk=Ashad./Awind;
vtj=vw1.*(1-sqrt(betatjtk.*(1-vw2./vw1).^2));

P_WT2=meshgrid(0.5.*rho.*A.*Cp2_opt*(vtj.^3));


% - WTs affected by multiple wake (0 WT)

% - Resultados

P_TOT_casD_no=7.*P_WT1+2.*P_WT2;

if max(P_WT1(:))>5000000
    P_WT1=5000000;
end

if max(P_WT2(:))>5000000
    P_WT2=5000000;
end

P_TOT_casD=7.*max(P_WT1(:))+2.*max(P_WT2(:));

[lambda1,lambda2]=meshgrid((2:0.01:9),(2:0.01:9));

[M_casD,I]=max(P_TOT_casD(:));
[I_row, I_col] = ind2sub(size(P_TOT_casD),I);
```

```
lambda2_final_casD=lambda2_opt;
lambda1_final_casD=lambda1(I_row,I_col);
Lambda2_final_casD=((lambda2_final_casD+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda2_final_casD+c8.*teta_pitch));
Cp2_final_casD=c1.*(c2.*(1./Lambda2_final_casD)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda2_final_casD));
Lambda1_final_casD=((lambda1_final_casD+c8.*teta_pitch).*(1+teta_pitch
.^3))./((1+teta_pitch.^3)-c9*(lambda1_final_casD+c8.*teta_pitch));
Cp1_final_casD=c1.*(c2.*(1./Lambda1_final_casD)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda1_final_casD));

casd2=M_casD;

end
```

- **tfg4_abcd2.m**

```
function casabcd = tfg4_abcd2(vwfree, alpha)

 %1. parametros de las WT
%2. Dibujar y graficar estelas y WT
%3. Matriz que clasifica wakes de cada WT
%4. Sin optimizar: Seg√∫n matriz, calcular velocidades que recibe cada
WT
%5. Calcular potencia optima


%% GENERIC CASE
%% 1.  WT parameters (adjust)

average = mean(wblrnd(8,2.26,[1 10000]));
normaldtr = (normrnd(average,0.5^2,[1 10000]));

rho=1.225; %air
R=63;
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003; %constants for equation Cp *
vw1=9.5;
teta_pitch=0; %blade pitch angle
lambda=2:0.01:9;
k=0.04;


lambda_optima=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);
lambda_optima_1decimal=round(lambda_optima,1);
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
```

```
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));
Cp_optima=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
Lambda=((lambda_optima+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_p
itch.^3)-c9*(lambda_optima+c8.*teta_pitch)); %como la A sin palito en
el medio
Cp_lambda_optima=c1.*(c2.*(1./Lambda)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda)); %USO MAS ADELANTE

%% 2. WTs coordenates ***9WT***

nwt=9; % **VARIABLE** numero de WT

alpha=alpha.*(pi./180);
beta=(pi./2)-alpha; %radians
xd=5.*D; %x distance between WT **VARIABLE**
yd=7.*D; %y distance between WT **VARIABLE**

vector_posicions_punt1=[0 0; xd 0; 2*xd 0; 0 -yd; xd -yd; 2*xd -yd; 0
-2*yd; xd -2*yd; 2*xd -2*yd]; %**VARIABLE**

scatter(vector_posicions_punt1(:,1),vector_posicions_punt1(:,2),'fille
d')
axis on
grid on
xlabel('x')
ylabel('y')
title('WIND TURBINES positions, case of 9 WT')
axis([-400 2200 -2200 400])
hold on %blau primer punt

%calculador del segon punt de la turbina
x_punt2=cos(alpha).*D;
y_punt2=sin(alpha).*D;

vector_posicions_punt2=vector_posicions_punt1+[x_punt2 y_punt2];

scatter(vector_posicions_punt2(:,1),vector_posicions_punt2(:,2),'fille
d','r')
axis on
grid on
xlabel('x')
ylabel('y')
title('WIND TURBINES positions, case of 9 WT')
axis([-400 2200 -2200 400])
legend('Primer punt posici√≥ turbina', 'Segon punt posici√≥ turbina')
%vermell segon punt

x=nwt*yd;
```

```
teta=atan(k);
e=(k.*x)./(sin(teta));


x_punt1k=sin(alpha-teta).*e;
y_punt1k=-cos(alpha-teta).*e;

x_punt2k=(sin(alpha+teta).*e);
y_punt2k=-(cos(alpha+teta).*e);

%calculador dels dos punts finals de la estela en generic
vector_posicions_punt1k=vector_posicions_punt1+[x_punt1k y_punt1k];
vector_posicions_punt2k=vector_posicions_punt2+[x_punt2k y_punt2k];

%calculador de estelas, resultat dos RECTES...
        %FORMA: y=M.*x + N, amb M i N dos matrius, cada fila es una
turbina
        %diferent, i 2 columnas, primera per punts 1&1k, segona per
2&2k:

M1_1k=(vector_posicions_punt1k(:,2)-
vector_posicions_punt1(:,2))./(vector_posicions_punt1k(:,1)-
vector_posicions_punt1(:,1));
M2_2k=(vector_posicions_punt2k(:,2)-
vector_posicions_punt2(:,2))./(vector_posicions_punt2k(:,1)-
vector_posicions_punt2(:,1));
M=[M1_1k,M2_2k];

N1_1k=vector_posicions_punt1k(:,2)-
M1_1k.*vector_posicions_punt1k(:,1);
N2_2k=vector_posicions_punt2k(:,2)-
M2_2k.*vector_posicions_punt2k(:,1);
N=[N1_1k,N2_2k];

scatter(vector_posicions_punt1k(:,1),vector_posicions_punt1k(:,2),'fil
led','g')
axis on
grid on
xlabel('x')
ylabel('y')
title('WIND TURBINES positions, case of 9 WT')
axis([-400 2200 -2200 400])
legend('Primer punt posici√≥ turbina', 'Segon punt posici√≥ turbina')
%verd tercer punt (estela)

scatter(vector_posicions_punt2k(:,1),vector_posicions_punt2k(:,2),'fil
led','g')
axis on
axis equal
grid on
xlabel('x')
ylabel('y')
title(['WIND TURBINES positions, case of 9 WT at ',
num2str(alpha*(180/pi)), ' degrees'])
axis([-3000 4800 -4800 3000])

%per dibuixar rectes esteles: (en vermell)
```

```
for e = 1:nwt
    plot([vector_posicions_punt1(e) vector_posicions_punt1k(e)],
[vector_posicions_punt1(e+nwt) vector_posicions_punt1k(e+nwt)], 'r')
    plot([vector_posicions_punt2(e) vector_posicions_punt2k(e)],
[vector_posicions_punt2(e+nwt) vector_posicions_punt2k(e+nwt)], 'r')
end

legend('Primer punt posici√≥ turbina', 'Segon punt posici√≥ turbina',
'Punts estela','Punts estela' ,'Forma esteles')
hold off

%% 3. Clasificacion de wakes
%definir una matriu que cada fila sigui el wake que reb la turbina n
%(n=fila=turbina que rep wake, m=columna=turbina que provoca wake)

matriu_wakes=strings(nwt);
turbines=[1:nwt]';
v_dir_vent=[sin(alpha) -cos(alpha)];
for esteles = 1:nwt
    for turbina = 1:nwt

        if turbina == esteles
            continue
        elseif (
(vector_posicions_punt1(turbina,1)>(vector_posicions_punt1(turbina,2)-
N(esteles,1))/M(esteles,1)) &&
(vector_posicions_punt2(turbina,1)<(vector_posicions_punt2(turbina,2)-
N(esteles,2))/M(esteles,2)) ) || (
(vector_posicions_punt1(turbina,1)<(vector_posicions_punt1(turbina,2)-
N(esteles,1))/M(esteles,1)) &&
(vector_posicions_punt2(turbina,1)>(vector_posicions_punt2(turbina,2)-
N(esteles,2))/M(esteles,2)) )

            AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("single wake");
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end
        elseif (
((vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) )  &&  (
((vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) )

            AeBe=[vector_posicions_punt1(turbina,1)-
```

```
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("partial wake");
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end

        elseif (
((vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) )  &&  (
((vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) )

            AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("partial wake");
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end

        else
            matriu_wakes(turbina,esteles)=strcat("");
        end
    end
end

if alpha==90.*(pi./180) || alpha==270.*(pi./180) ||
alpha==91.*(pi./180) || alpha==271.*(pi./180) || alpha==92.*(pi./180)
|| alpha==272.*(pi./180) || alpha==89.*(pi./180) ||
alpha==269.*(pi./180) || alpha==88.*(pi./180) || alpha==268.*(pi./180)
    matriu_wakes=strings(nwt);
    for esteles = 1:nwt
        for turbina = 1:nwt
            if turbina == esteles
                continue %break
            elseif  (
(vector_posicions_punt1(turbina,2)>(M(esteles,1)*vector_posicions_punt
1(turbina,1)+N(esteles,1))) &&
(vector_posicions_punt2(turbina,2)<(M(esteles,2)*vector_posicions_punt
2(turbina,1)+N(esteles,2))) ) || (
(vector_posicions_punt1(turbina,2)<(M(esteles,1)*vector_posicions_punt
1(turbina,1)+N(esteles,1))) &&
(vector_posicions_punt2(turbina,2)>(M(esteles,2)*vector_posicions_punt
2(turbina,1)+N(esteles,2))) )
```

```
                AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
                if dot(AeBe,v_dir_vent)>0
                    matriu_wakes(turbina,esteles)=strcat("single
wake");
                else
                    matriu_wakes(turbina,esteles)=strcat("");
                end
            end
        end
    end
end


%% 4. Velocidades que recibe cada WT, seg√∫n matriz (TIPOS DE WAKES)

vwfree=vwfree; %[m/s] speed free stream **VARIABLE**
vector_velocitats=zeros(nwt,1);


%1. LOOP for free stream turbines
for fila = 1:nwt
    turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
    contador=0;
    if numel(matriu_wakes(turbina_fila=="single wake"))==0 &&
numel(matriu_wakes(turbina_fila=="partial wake"))==0
       vector_velocitats(fila)=vwfree;
    end
end


while size(find(vector_velocitats==0),1)>0

    %2. LOOP for single wake (1 o mas) EXCLUSIVO
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="single wake"))>0 &&
numel(matriu_wakes(turbina_fila=="partial wake"))==0
            %EXCLUSIVO SINGLE WAKE
            [columna]=find(turbina_fila=="single wake");

            if size(columna)==[1 1]
               AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna,1) vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna,2)];
               xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
               vector_velocitats(fila)=vector_velocitats(columna).*(1-
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));
            else %LOOP for +1 SingleS wakes
                num_esteles_single=size(columna);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(1),1) vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2)); %inicialitzem amb primer valor de esteles
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
```

```
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt;
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vector_velocitats(fila)=vector_velocitats(columna(element_xwt_menor)).
*(1-((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

            end
        end
    end


    %3. LOOP for partial wake EXCLUSIVO (solo 1 partial)
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))>0 &&
numel(matriu_wakes(turbina_fila=="single wake"))==0
            [columna]=find(turbina_fila=="partial wake");

            if size(columna)==[1 1]

               xq=x.*sin(alpha)+vector_posicions_punt1(columna,1);
               yq=-x.*cos(alpha)+vector_posicions_punt1(columna,2);%OK
               vvx=xq-vector_posicions_punt1(columna,1);
               vvy=yq-vector_posicions_punt1(columna,2);%OK
               vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
               vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

               k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna,2)));
               lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna,1)).*k_vector))./(vector_posicions_punt
2(fila,1)-vector_posicions_punt1(fila,1));

               xa=xq+k_vector.*(xq-vector_posicions_punt1(columna,1));
```

```
                ya=yq+k_vector.*(yq-vector_posicions_punt1(columna,2));

                x_a_single=sqrt(((xa-
vector_posicions_punt1(columna,1)).^2)+((ya-
vector_posicions_punt1(columna,2)).^2));
                vw_a_single=vector_velocitats(columna).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));


                Dx=D+2.*k.*x_a_single;
                r2=R;
                r1=Dx./2;
                d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                Awind=pi.*(r2.^2);
                Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                betatjtk=Ashad./Awind;
                vector_velocitats(fila)=vector_velocitats(columna).*(1-
sqrt(betatjtk.*(1-vw_a_single./vector_velocitats(columna)).^2));

              end
          end
      end


      %4. LOOP for partial wake EXCLUSIVO (multiple partial wakes)
for fila = 1:nwt
          turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
          contador=0;
          if numel(matriu_wakes(turbina_fila=="partial wake"))>0 &&
numel(matriu_wakes(turbina_fila=="single wake"))==0
              [columna]=find(turbina_fila=="partial wake");

              if size(columna)>[0 1]
                  sumes_arrel=0;
                  num_esteles_partial=size(columna);
                  for element = 1:num_esteles_partial(2)


xq=x.*sin(alpha)+vector_posicions_punt1(columna(element),1);
                      yq=-
x.*cos(alpha)+vector_posicions_punt1(columna(element),2);%OK
                      vvx=xq-vector_posicions_punt1(columna(element),1);
                      vvy=yq-
vector_posicions_punt1(columna(element),2);%OK
                      vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                      vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                      k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
```

```
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna(element),1)))./(vector_posicions_punt2(
fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna(element),1)).*k_vector))./(vector_posic
ions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna(element),1));

                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna(element),1)).^2)+((ya-
vector_posicions_punt1(columna(element),2)).^2));

vw_a_single=vector_velocitats(columna(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna(element)).*(1-sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end

                vector_velocitats(fila)=vwfree.*(1-sqrt(sumes_arrel));
            end
        end
    end


    %5. LOOP for single WAKE (1 o mas) + 1 solo partial wake
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))==1 &&
numel(matriu_wakes(turbina_fila=="single wake"))>0
            [columna_partial]=find(turbina_fila=="partial wake");
            [columna_single]=find(turbina_fila=="partial wake");

            if size(columna_single)==[1 1]
                %single:
```

```
                AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single,1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single,2)];
                xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                vw_single=vector_velocitats(columna_single).*(1-
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                %partial:

xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial,1);
                yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial,2);%OK
                vvx=xq-vector_posicions_punt1(columna_partial,1);
                vvy=yq-vector_posicions_punt1(columna_partial,2);%OK
                vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial,1)))./(vector_posicions_punt2(f
ila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial,2)));
                lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial,1)).*k_vector))./(vector_posici
ons_punt2(fila,1)-vector_posicions_punt1(fila,1));
                xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial,1));

                ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial,2));

                x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial,1)).^2)+((ya-
vector_posicions_punt1(columna_partial,2)).^2));
                vw_a_single=vector_velocitats(columna_partial).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                Dx=D+2.*k.*x_a_single;
                r2=R;
                r1=Dx./2;
                d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                Awind=pi.*(r2.^2);
                Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                betatjtk=Ashad./Awind;
                vw_partial=vector_velocitats(columna_partial).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial)).^2));

                %multiple:
                vector_velocitats(fila)=vwfree.*(1-sqrt(((1-
vw_partial./vwfree).^2)+((1-vw_single./vwfree).^2)));
```

```
            else
                %single:
                num_esteles_single=size(columna_single);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(1),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2));
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt; %pren valor menor
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vw_single=vector_velocitats(columna_single(element_xwt_menor)).*(1-
((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                %partial:

xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial,1);
                yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial,2);%OK
                vvx=xq-vector_posicions_punt1(columna_partial,1);
                vvy=yq-vector_posicions_punt1(columna_partial,2);%OK
                vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial,1)))./(vector_posicions_punt2(f
ila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial,2)));
                lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial,1)).*k_vector))./(vector_posici
ons_punt2(fila,1)-vector_posicions_punt1(fila,1));
                xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial,1));
```

ETSEIB

```
                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial,2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial,1)).^2)+((ya-
vector_posicions_punt1(columna_partial,2)).^2));
                    vw_a_single=vector_velocitats(columna_partial).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;
                    vw_partial=vector_velocitats(columna_partial).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial)).^2));

                    %multiple:
                    vector_velocitats(fila)=vwfree.*(1-sqrt(((1-
vw_partial./vwfree).^2)+((1-vw_single./vwfree).^2)));

               end
          end
     end


     %6. LOOP for single WAKE (1 o mas) + >1 partial wake
     for fila = 1:nwt
          turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
          contador=0;
          if numel(matriu_wakes(turbina_fila=="partial wake"))>1 &&
numel(matriu_wakes(turbina_fila=="single wake"))>0
               [columna_partial]=find(turbina_fila=="partial wake");
               [columna_single]=find(turbina_fila=="partial wake");

               if size(columna_single)==[1 1] %LOOP para >1 partial i 1
single
                    %single:
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single,1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single,2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    vw_single=vector_velocitats(columna_single).*(1-
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    %partial:
                    sumes_arrel=0;
                    num_esteles_partial=size(columna_partial); %numero de
esteles que afecten partial
                    for element = 1:num_esteles_partial(2)
```

```
xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial(element),1);
                    yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial(element),2);%OK
                    vvx=xq-
vector_posicions_punt1(columna_partial(element),1);
                    vvy=yq-
vector_posicions_punt1(columna_partial(element),2);%OK
                    vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                    vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);
                    k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial(element),1)))./(vector_posicion
s_punt2(fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial(element),1)).*k_vector))./(vect
or_posicions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial(element),1));

                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial(element),1)).^2)+((ya-
vector_posicions_punt1(columna_partial(element),2)).^2)); %distance
vw_a_single=vector_velocitats(columna_partial(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna_partial(element)).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end
```

```
                vector_velocitats(fila)=vwfree.*(1-
sqrt(sumes_arrel+((1-vw_single./vwfree).^2)));

            else %LOOP for +1 SingleS wakes y >1 partial
                %single:
                num_esteles_single=size(columna_single);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(1),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2)); %inicialitzem amb primer valor de esteles
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt; %pren valor menor
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vw_single=vector_velocitats(columna_single(element_xwt_menor)).*(1-
((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                %partial:
                sumes_arrel=0;
                num_esteles_partial=size(columna_partial);
                for element = 1:num_esteles_partial(2)

                    %columna(element)


xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial(element),1);
                    yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial(element),2);%OK
                    vvx=xq-
vector_posicions_punt1(columna_partial(element),1);
                    vvy=yq-
vector_posicions_punt1(columna_partial(element),2);%OK
                    vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                    vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                    k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
```

```
vector_posicions_punt1(fila,1)))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial(element),1)))./(vector_posicion
s_punt2(fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial(element),1)).*k_vector))./(vect
or_posicions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial(element),1));

                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial(element),1)).^2)+((ya-
vector_posicions_punt1(columna_partial(element),2)).^2));
vw_a_single=vector_velocitats(columna_partial(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna_partial(element)).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end

                vector_velocitats(fila)=vwfree.*(1-
sqrt(sumes_arrel+((1-vw_single./vwfree).^2))); %MULTIPLE PARTIAL WAKE
EQUATION

            end
        end
    end


end


%% 5. Calcular potencia optima

potencia_total=0;
for fila = 1:nwt
```

```
    if
(0.5.*Cp_lambda_optima.*rho.*A.*(vector_velocitats(fila))^3)>5000000
        potencia_total=potencia_total+5000000;
    else

potencia_total=potencia_total+(0.5.*Cp_lambda_optima.*rho.*A.*(vector_
velocitats(fila))^3);
    end
end


%vector_velocitats;
casabcd = potencia_total;

end
```

- **tfg3_calling.m**

```
%function calling for TFG3

%% 9 WT case
clear all;clc;
close('all')
format long

rho=1.225; %air
R=63; %D/2 minus hub (blades length) *TMdPG1de1.pdf
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003;
teta_pitch=0; %blade pitch angle
lambda=2:0.01:9;

%% case A: 0, 180 degrees

vwfree=3:25;
vwlength=length(vwfree);
vector_power=zeros([1 vwlength]);
vector_power2=zeros([1 vwlength]);

for element=1:vwlength
    vw1=vwfree(element);
    vector_power(element)=tfg3_a(vw1);
end

for element2=1:vwlength
    vw1=vwfree(element2);
    vector_power2(element2)=tfg3_a2(vw1);
end

%%
plot(3:25,vector_power/1000000,  'g', 'LineWidth', 2)
axis on
%axis equal
xlabel('Wind speed [m/s]')
ylabel('Power [MW]')
```

```
title(['Total power generated for wind direction of 0 and 180
degrees'])
axis([0 25 0 100])
hold on
plot(3:25, vector_power2/1000000, 'b', 'LineWidth', 2)
legend('Proposed WPP operation', 'Conventional WPP operation')



%% case B: 90, 270 degrees

vwfree=3:25;
vwlength=length(vwfree);
vector_power=zeros([1 vwlength]);
vector_power2=zeros([1 vwlength]);

for element=1:vwlength
    vw1=vwfree(element);
    vector_power(element)=tfg3_b(vw1);
end

for element2=1:vwlength
    vw1=vwfree(element2);
    vector_power2(element2)=tfg3_b2(vw1);
end

%%
plot(3:25,vector_power/1000000,  'g', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind speed [m/s]')
ylabel('Power [MW]')
title(['Total power generated for wind direction of 90 and 270
degrees'])
axis([0 25 0 100])
hold on
plot(3:25, vector_power2/1000000, 'b', 'LineWidth', 2)
legend('Proposed WPP operation', 'Conventional WPP operation')

%% case C: 30, 150, 210 and 330 (casC)

vwfree_vec=3:25;
vwlength=length(vwfree_vec);
vector_power=zeros([1 vwlength]);
vector_power2=zeros([1 vwlength]);

for element=1:vwlength
    vw1=vwfree_vec(element);
    vector_power(element)=tfg3_c(vw1);
end

for element2=1:vwlength
    vwfree=vwfree_vec(element2);
    vector_power2(element2)=tfg4_abcd2(vwfree, 30);
end

%%
```

```
plot(3:25, vector_power2/1000000, 'g', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind speed [m/s]')
ylabel('Power [MW]')
title(['Total power generated for wind direction of 30, 150, 210 and
330 degrees'])
axis([0 25 0 100])
hold on
plot(3:25,vector_power/1000000,  'b', 'LineWidth', 2)
legend('Proposed WPP operation', 'Conventional WPP operation')


%% case D 60, 120, 240 and 300 degrees


vwfree=3:25;
vwlength=length(vwfree);
vector_power=zeros([1 vwlength]);
vector_power2=zeros([1 vwlength]);

for element=1:vwlength
    vw1=vwfree(element);
    vector_power(element)=tfg3_d(vw1);
end

for element2=1:vwlength
    vw1=vwfree(element2);
    vector_power2(element2)=tfg3_d2(vw1);
end

%%
plot(3:25,vector_power/1000000,  'g', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind speed [m/s]')
ylabel('Power [MW]')
title(['Total power generated for wind direction of 60, 120, 240 and
300 degrees'])
axis([0 25 0 100])
hold on
plot(3:25, vector_power2/1000000, 'b', 'LineWidth', 2)
legend('Proposed WPP operation', 'Conventional WPP operation')
```

4. Forth section: Fécamp generic case

- **fecamp_function.m**

```
function total_power = fecamp_function(vwfree,alpha)


clc; %clear all;
```

```
format long
close('all')

%% Campo eolico de Fecamp

%extraer posiciones turbinas de la figura 'fecamp layout.fig'
%tener la .fig guardada en la misma carpeta que este archivo .m

fig = openfig('fecamp layout.fig');
figure()

dataObjs = findobj(fig,'-property','YData');
dataObjs = findobj(fig,'-property','XData');

x_fecamp = dataObjs(85).XData;
y_fecamp = dataObjs(85).YData;
%85 ja que hi ha 83 turbines mes el punt central OSS, seguent 85

%close(fig)

%% 1.  WT parameters (adjust)

average = mean(wblrnd(8,2.26,[1 10000]));
normaldtr = (normrnd(average,0.5^2,[1 10000]));

rho=1.225; %air
R=75; %D/2 minus hub (blades length) *TMdPG1de1.pdf **VARIABLE**
D=R*2;
A=pi.*(R^2); %swept area
c1=0.73; c2=151; c3=0.58; c4=0.002; c5=2.14; c6=13.2; c7=18.4; c8=-
0.02; c9=-0.003; %constants for equation Cp *
vw1=9.5;
teta_pitch=0; %blade pitch
lambda=2:0.01:9;
k=0.04; %Jensen estela para offshore applications

lambda_optima=(c2.*c7)./(c2.*c9.*c7+c6.*c7+c2);
lambda_optima_1decimal=round(lambda_optima,1);
Ct_vector=[0.121803150380010,0.125991426879747,0.130372866788154,0.134
947794202068,0.139716548938385,0.144679480924098,0.149836945852410,0.1
55189301827121,0.160736906779912,0.166480116493183,0.172419283098105,0
.178554753945989,0.184886870773109,0.191415969096183,0.198142377789023
,0.205122378008047,0.212363973950428,0.219884823744599,0.2277165741478
05,0.235905752851484,0.244486035989457,0.253486104334996,0.26291155794
3879,0.272757594242936,0.283007408104841,0.293646835724271,0.304653069
492279,0.316010028258569,0.327696636021499,0.339693939413586,0.3519874
52539419,0.364556380282990,0.377383832357742,0.390456906615975,0.40375
2894555614,0.417266029429188,0.430970459886915,0.444858279381295,0.458
909231955795,0.473108898876679,0.487439791375504,0.501882662099191,0.5
16427401248424,0.531048382940466,0.545703059988886,0.560416750539903,0
.575063029023819,0.589679761955707,0.602954103820260,0.617061697842447
,0.630471356340617,0.643136907473184,0.654967413059928,0.6668359762164
01,0.678127641069006,0.689303051544152,0.700396551059694,0.71140722733
5882,0.722307736139963,0.733079282358001,0.743634250978271,0.754095088
731837,0.764282533105000,0.774264611585461,0.784142619928012,0.7936657
00483269,0.802933289131714,0.812123022351999,0.820863428582445,0.82931
4455736052,0.837463579391426;];
lambda_Ct=2:0.1:9;
Ct_lambda_optima=Ct_vector(find(lambda_Ct==lambda_optima_1decimal));
```

```
Cp_optima=((c1.*c2)./c7).*exp(((-c6.*c7)./c2)-1);
Lambda=((lambda_optima+c8.*teta_pitch).*(1+teta_pitch.^3))./((1+teta_p
itch.^3)-c9*(lambda_optima+c8.*teta_pitch));
Cp_lambda_optima=c1.*(c2.*(1./Lambda)-c3.*teta_pitch-
c4.*(teta_pitch.^c5)-c6).*exp(-c7.*(1./Lambda)); %USO MAS ADELANTE


%% 2. WTs coordenates ***9WT***

nwt=83; % **VARIABLE** numero de WT -OK

%disposicions cas 0 graus!!:
alpha=alpha.*(pi./180);
beta=(pi./2)-alpha; %radians


%Borro el punt OSS
x_fecamp=x_fecamp(1:83);
y_fecamp=y_fecamp(1:83);

vector_posicions_punt1=horzcat(x_fecamp',y_fecamp');

names=[1:83]'; b_names = num2str(names); c_names = cellstr(b_names);
dx_names = 80; dy_names = 80;

scatter(vector_posicions_punt1(:,1),vector_posicions_punt1(:,2),'fille
d', 'black')
axis on
axis equal
grid on
text(vector_posicions_punt1(:,1)+dx_names,vector_posicions_punt1(:,2)+
dy_names, c_names, 'Fontsize', 8);
xlabel('x')
ylabel('y')
title('WIND TURBINES positions of F√©camp')
axis([0 12000 0 15000])
hold on %negre primer punt

%calculador del segon punt de la turbina
x_punt2=cos(alpha).*D;
y_punt2=sin(alpha).*D;

vector_posicions_punt2=vector_posicions_punt1+[x_punt2 y_punt2];

scatter(vector_posicions_punt2(:,1),vector_posicions_punt2(:,2),'fille
d','r')
axis on
axis equal
grid on
text(vector_posicions_punt1(:,1)+dx_names,vector_posicions_punt1(:,2)+
dy_names, c_names, 'Fontsize', 8);
xlabel('x')
ylabel('y')
title('WIND TURBINES positions of F√©camp')
axis([0 12000 0 15000])
legend('Primer punt posici√≥ turbina', 'Segon punt posici√≥ turbina')
%vermell segon punt

xd=1169; %excel fecamp
```

```
x=nwt*xd;

%teta=10.*(pi./180);
teta=atan(k);
e=(k.*x)./(sin(teta));

%fet a ma, calculs fets a la llibreta
x_punt1k=sin(alpha-teta).*e;
y_punt1k=-cos(alpha-teta).*e;

x_punt2k=(sin(alpha+teta).*e);
y_punt2k=-(cos(alpha+teta).*e);

%calculador dels dos punts finals de la estela en generic
vector_posicions_punt1k=vector_posicions_punt1+[x_punt1k y_punt1k];
vector_posicions_punt2k=vector_posicions_punt2+[x_punt2k y_punt2k];

M1_1k=(vector_posicions_punt1k(:,2)-
vector_posicions_punt1(:,2))./(vector_posicions_punt1k(:,1)-
vector_posicions_punt1(:,1));
M2_2k=(vector_posicions_punt2k(:,2)-
vector_posicions_punt2(:,2))./(vector_posicions_punt2k(:,1)-
vector_posicions_punt2(:,1));
M=[M1_1k,M2_2k];

N1_1k=vector_posicions_punt1k(:,2)-
M1_1k.*vector_posicions_punt1k(:,1);
N2_2k=vector_posicions_punt2k(:,2)-
M2_2k.*vector_posicions_punt2k(:,1);
N=[N1_1k,N2_2k];

scatter(vector_posicions_punt1k(:,1),vector_posicions_punt1k(:,2),'fil
led','g')
axis on
axis equal
grid on
text(vector_posicions_punt1(:,1)+dx_names,vector_posicions_punt1(:,2)+
dy_names, c_names, 'Fontsize', 8);
xlabel('x')
ylabel('y')
title('WIND TURBINES positions of F√©camp')
axis([0 12000 0 15000])
legend('Primer punt posici√≥ turbina', 'Segon punt posici√≥ turbina')
%verd tercer punt (estela)

scatter(vector_posicions_punt2k(:,1),vector_posicions_punt2k(:,2),'fil
led','g')
axis on
axis equal
grid on
text(vector_posicions_punt1(:,1)+dx_names,vector_posicions_punt1(:,2)+
dy_names, c_names, 'Fontsize', 8);
xlabel('x')
ylabel('y')
title(['WIND TURBINES positions of F√©camp, wind turbines disposition
at ', num2str(alpha*(180/pi)), ' degrees'])
axis([0 12000 0 15000])

%per dibuixar rectes esteles: (en vermell)
```

```
for e = 1:nwt
    plot([vector_posicions_punt1(e) vector_posicions_punt1k(e)],
[vector_posicions_punt1(e+nwt) vector_posicions_punt1k(e+nwt)], 'r')
    plot([vector_posicions_punt2(e) vector_posicions_punt2k(e)],
[vector_posicions_punt2(e+nwt) vector_posicions_punt2k(e+nwt)], 'r')
end


legend('First WT point position', 'Second WT point position', 'Wake
points','Wake points' ,'Wake lines', 'Location','northeastoutside')
hold off

%% 3. Clasificacion de wakes
matriu_wakes=strings(nwt);
turbines=[1:nwt]';
v_dir_vent=[sin(alpha) -cos(alpha)]; %pel producte escalar, te sentit
i direccio

%!!! turbina(#fila)=turbina rep wake de la turbina(#columna)=esteles

for esteles = 1:nwt
    for turbina = 1:nwt

        if turbina == esteles
            continue %es como break
        elseif (
(vector_posicions_punt1(turbina,1)>(vector_posicions_punt1(turbina,2)-
N(esteles,1))/M(esteles,1)) &&
(vector_posicions_punt2(turbina,1)<(vector_posicions_punt2(turbina,2)-
N(esteles,2))/M(esteles,2)) ) || (
(vector_posicions_punt1(turbina,1)<(vector_posicions_punt1(turbina,2)-
N(esteles,1))/M(esteles,1)) &&
(vector_posicions_punt2(turbina,1)>(vector_posicions_punt2(turbina,2)-
N(esteles,2))/M(esteles,2)) )

            AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("single wake");
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end
        elseif (
((vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) )  &&  (
((vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2)
```

```
-N(esteles,2))/M(esteles,2)))) )
            %matriu_wakes(turbina,esteles)=strcat("partial wake");
            AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("partial wake");
%from WT",num2str(esteles)); (anterior)
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end

        elseif (
((vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt2(turbina,1)>((vector_posicions_punt2(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt2(turbina,1)<((vector_posicions_punt2(turbina,2)
-N(esteles,2))/M(esteles,2)))) )  &&  (
((vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)<((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) ||
((vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2
)-N(esteles,1))/M(esteles,1))) &&
(vector_posicions_punt1(turbina,1)>((vector_posicions_punt1(turbina,2)
-N(esteles,2))/M(esteles,2)))) )
            %matriu_wakes(turbina,esteles)=strcat("partial wake");
            AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
            if dot(AeBe,v_dir_vent)>0
                matriu_wakes(turbina,esteles)=strcat("partial wake");
%from WT",num2str(esteles)); (anterior)
            else
                matriu_wakes(turbina,esteles)=strcat("");
            end

        else
            matriu_wakes(turbina,esteles)=strcat("");
        end
    end
end

if alpha==90.*(pi./180) || alpha==270.*(pi./180) ||
alpha==91.*(pi./180) || alpha==271.*(pi./180) || alpha==92.*(pi./180)
|| alpha==272.*(pi./180) || alpha==89.*(pi./180) ||
alpha==269.*(pi./180) || alpha==88.*(pi./180) || alpha==268.*(pi./180)
    matriu_wakes=strings(nwt);
    for esteles = 1:nwt
        for turbina = 1:nwt
            if turbina == esteles
                continue %es como break
            elseif  (
(vector_posicions_punt1(turbina,2)>(M(esteles,1)*vector_posicions_punt
1(turbina,1)+N(esteles,1))) &&
(vector_posicions_punt2(turbina,2)<(M(esteles,2)*vector_posicions_punt
2(turbina,1)+N(esteles,2))) ) || (
(vector_posicions_punt1(turbina,2)<(M(esteles,1)*vector_posicions_punt
```

```
1(turbina,1)+N(esteles,1))) &&
(vector_posicions_punt2(turbina,2)>(M(esteles,2)*vector_posicions_punt
2(turbina,1)+N(esteles,2))) )

                AeBe=[vector_posicions_punt1(turbina,1)-
vector_posicions_punt1(esteles,1) vector_posicions_punt1(turbina,2)-
vector_posicions_punt1(esteles,2)];
                if dot(AeBe,v_dir_vent)>0
                    matriu_wakes(turbina,esteles)=strcat("single
wake");
                else
                    matriu_wakes(turbina,esteles)=strcat("");
                end
            end
        end
    end
end



%% 4. Velocidades que recibe cada WT, seg√∫n matriz (TIPOS DE WAKES)

vector_velocitats=zeros(nwt,1);

%!!! turbina(#fila)=turbina rep wake de la turbina(#columna)=esteles

%cut out speed of 25m/s

%1. LOOP for free stream turbines
for fila = 1:nwt
    turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
    contador=0;
    if numel(matriu_wakes(turbina_fila=="single wake"))==0 &&
numel(matriu_wakes(turbina_fila=="partial wake"))==0
        if vwfree>25
           vector_velocitats(fila)=0; %cut out wind speed
        else
           vector_velocitats(fila)=vwfree;
        end
    end
end

while size(find(vector_velocitats==0),1)>0 %&&

    %2. LOOP for single wake (1 o mas) EXCLUSIVO
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="single wake"))>0 &&
numel(matriu_wakes(turbina_fila=="partial wake"))==0

            [columna]=find(turbina_fila=="single wake");

            if size(columna)==[1 1]
               AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna,1) vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna,2)];
               xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
               vector_velocitats(fila)=vector_velocitats(columna).*(1-
```

```
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));
            else %LOOP for +1 SingleS wakes
                num_esteles_single=size(columna);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(1),1) vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2));
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt; %pren valor menor
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vector_velocitats(fila)=vector_velocitats(columna(element_xwt_menor)).
*(1-((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

            end
        end
    end


    %3. LOOP for partial wake EXCLUSIVO (solo 1 partial)
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))>0 &&
numel(matriu_wakes(turbina_fila=="single wake"))==0
            [columna]=find(turbina_fila=="partial wake");

            if size(columna)==[1 1]

               xq=x.*sin(alpha)+vector_posicions_punt1(columna,1);
               yq=-x.*cos(alpha)+vector_posicions_punt1(columna,2);%OK
               vvx=xq-vector_posicions_punt1(columna,1);
               vvy=yq-vector_posicions_punt1(columna,2);%OK
               vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
               vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2
               k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1)))))./((((vector_posicions_punt2(fila,2)-
```

```
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna,2)));
               lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna,1)).*k_vector))./(vector_posicions_punt
2(fila,1)-vector_posicions_punt1(fila,1));

               xa=xq+k_vector.*(xq-vector_posicions_punt1(columna,1));

               ya=yq+k_vector.*(yq-vector_posicions_punt1(columna,2));

               x_a_single=sqrt(((xa-
vector_posicions_punt1(columna,1)).^2)+((ya-
vector_posicions_punt1(columna,2)).^2));
               vw_a_single=vector_velocitats(columna).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));


               Dx=D+2.*k.*x_a_single;
               r2=R;
               r1=Dx./2;
               d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
               Awind=pi.*(r2.^2);
               Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
               betatjtk=Ashad./Awind;
               vector_velocitats(fila)=vector_velocitats(columna).*(1-
sqrt(betatjtk.*(1-vw_a_single./vector_velocitats(columna)).^2));

            end
        end
    end


    %4. LOOP for partial wake EXCLUSIVO (multiple partial wakes)

    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))>0 &&
numel(matriu_wakes(turbina_fila=="single wake"))==0
            [columna]=find(turbina_fila=="partial wake");

            if size(columna)>[0 1]
                sumes_arrel=0;
                num_esteles_partial=size(columna);
                for element = 1:num_esteles_partial(2)

                    %columna(element)


xq=x.*sin(alpha)+vector_posicions_punt1(columna(element),1);
                    yq=-
x.*cos(alpha)+vector_posicions_punt1(columna(element),2);%OK
```

```
                    vvx=xq-vector_posicions_punt1(columna(element),1);
                    vvy=yq-
vector_posicions_punt1(columna(element),2);%OK
                    vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                    vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                    k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna(element),1)))./(vector_posicions_punt2(
fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna(element),1)).*k_vector))./(vector_posic
ions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna(element),1));

                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna(element),1)).^2)+((ya-
vector_posicions_punt1(columna(element),2)).^2));

vw_a_single=vector_velocitats(columna(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna(element)).*(1-sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end

                vector_velocitats(fila)=vwfree.*(1-sqrt(sumes_arrel));
%MULTIPLE PARTIAL WAKE EQUATION
            end
        end
    end
```

```
    %5. LOOP for single WAKE (1 o mas) + 1 solo partial wake
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))==1 &&
numel(matriu_wakes(turbina_fila=="single wake"))>0
            [columna_partial]=find(turbina_fila=="partial wake");
            [columna_single]=find(turbina_fila=="partial wake");

            if size(columna_single)==[1 1]
                %single:
                AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single,1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single,2)];
                xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                vw_single=vector_velocitats(columna_single).*(1-
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                %partial:

xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial,1);
                yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial,2);%OK
                vvx=xq-vector_posicions_punt1(columna_partial,1);
                vvy=yq-vector_posicions_punt1(columna_partial,2);%OK
                vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial,1)))./(vector_posicions_punt2(f
ila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial,2)));
                lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial,1)).*k_vector))./(vector_posici
ons_punt2(fila,1)-vector_posicions_punt1(fila,1));
                xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial,1));

                ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial,2));

                x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial,1)).^2)+((ya-
vector_posicions_punt1(columna_partial,2)).^2));
                vw_a_single=vector_velocitats(columna_partial).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));
%average wind speed at hub height of WT2 with wake effect, Jensen
single wake
                Dx=D+2.*k.*x_a_single;
                r2=R;
                r1=Dx./2;
```

```
                d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                Awind=pi.*(r2.^2);
                Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                betatjtk=Ashad./Awind;
                vw_partial=vector_velocitats(columna_partial).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial)).^2));

                %multiple:
                vector_velocitats(fila)=vwfree.*(1-sqrt(((1-
vw_partial./vwfree).^2)+((1-vw_single./vwfree).^2)));

            else
                %single:
                num_esteles_single=size(columna_single);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(1),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2));
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt;
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vw_single=vector_velocitats(columna_single(element_xwt_menor)).*(1-
((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                %partial:

xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial,1);
                yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial,2);%OK
                vvx=xq-vector_posicions_punt1(columna_partial,1);
                vvy=yq-vector_posicions_punt1(columna_partial,2);%OK
                vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK
```

```
                k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial,1)))./(vector_posicions_punt2(f
ila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial,2)));
                lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial,1)).*k_vector))./(vector_posici
ons_punt2(fila,1)-vector_posicions_punt1(fila,1));
                xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial,1));

                ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial,2));
                x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial,1)).^2)+((ya-
vector_posicions_punt1(columna_partial,2)).^2));
                vw_a_single=vector_velocitats(columna_partial).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                Dx=D+2.*k.*x_a_single;
                r2=R;
                r1=Dx./2;
                d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                betatjtk=Ashad./Awind;
                vw_partial=vector_velocitats(columna_partial).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial)).^2));

                %multiple:
                vector_velocitats(fila)=vwfree.*(1-sqrt(((1-
vw_partial./vwfree).^2)+((1-vw_single./vwfree).^2)));

            end
        end
    end


    %6. LOOP for single WAKE (1 o mas) + >1 partial wake
    for fila = 1:nwt
        turbina_fila=matriu_wakes(fila,:); %matriz [1xnwt]
        contador=0;
        if numel(matriu_wakes(turbina_fila=="partial wake"))>1 &&
numel(matriu_wakes(turbina_fila=="single wake"))>0
            [columna_partial]=find(turbina_fila=="partial wake");
            [columna_single]=find(turbina_fila=="partial wake");

            if size(columna_single)==[1 1] %LOOP para >1 partial i 1
single
                %single:
                AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single,1)
```

```
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single,2)];
                xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                vw_single=vector_velocitats(columna_single).*(1-
((D./(D+2.*k.*xwt)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                %partial:
                sumes_arrel=0;
                num_esteles_partial=size(columna_partial);
                for element = 1:num_esteles_partial(2)

                    %columna(element)


xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial(element),1);
                    yq=-
x.*cos(alpha)+vector_posicions_punt1(columna_partial(element),2);%OK
                    vvx=xq-
vector_posicions_punt1(columna_partial(element),1);
                    vvy=yq-
vector_posicions_punt1(columna_partial(element),2);%OK
                    vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                    vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                    k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial(element),1)))./(vector_posicion
s_punt2(fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial(element),1)).*k_vector))./(vect
or_posicions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial(element),1));

                    ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial(element),1)).^2)+((ya-
vector_posicions_punt1(columna_partial(element),2)).^2));

vw_a_single=vector_velocitats(columna_partial(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));
                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
```

```
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna_partial(element)).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end

                vector_velocitats(fila)=vwfree.*(1-
sqrt(sumes_arrel+((1-vw_single./vwfree).^2))); %MULTIPLE PARTIAL WAKE
EQUATION

            else %LOOP for +1 SingleS wakes y >1 partial
                %single:
                num_esteles_single=size(columna_single);
                AeBe_primera_estela=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(1),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(1),2)];

xwt_primera_estela=sqrt(((AeBe_primera_estela(1)).^2)+((AeBe_primera_e
stela(2)).^2)); %inicialitzem amb primer valor de esteles
                element_xwt_menor=1;
                for element = 2:num_esteles_single(2)
                    AeBe=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element),2)];
                    xwt=sqrt(((AeBe(1)).^2)+((AeBe(2)).^2));
                    if xwt < xwt_primera_estela
                        xwt_primera_estela = xwt;
                        element_xwt_menor=element;
                    else
                        continue
                    end
                end
                AeBe_def=[vector_posicions_punt1(fila,1)-
vector_posicions_punt1(columna_single(element_xwt_menor),1)
vector_posicions_punt1(fila,2)-
vector_posicions_punt1(columna_single(element_xwt_menor),2)];
                xwt_def=sqrt(((AeBe_def(1)).^2)+((AeBe_def(2)).^2));

vw_single=vector_velocitats(columna_single(element_xwt_menor)).*(1-
((D./(D+2.*k.*xwt_def)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                %partial:
                sumes_arrel=0;
                num_esteles_partial=size(columna_partial);
                for element = 1:num_esteles_partial(2)

                    %columna(element)


xq=x.*sin(alpha)+vector_posicions_punt1(columna_partial(element),1);
                    yq=-
```

ETSEIB

```
x.*cos(alpha)+vector_posicions_punt1(columna_partial(element),2);%OK
                    vvx=xq-
vector_posicions_punt1(columna_partial(element),1);
                    vvy=yq-
vector_posicions_punt1(columna_partial(element),2);%OK
                    vtx=vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1);
                    vty=vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2);%OK

                    k_vector=((yq-vector_posicions_punt2(fila,2))-
(((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt2(fila,1)))./(vector_posicions_punt2(fila,1)-
vector_posicions_punt1(fila,1))))./((((vector_posicions_punt2(fila,2)-
vector_posicions_punt1(fila,2)).*(xq-
vector_posicions_punt1(columna_partial(element),1)))./(vector_posicion
s_punt2(fila,1)-vector_posicions_punt1(fila,1)))-(yq-
vector_posicions_punt1(columna_partial(element),2)));
                    lambda_vector=((xq-
vector_posicions_punt2(fila,1))+((xq-
vector_posicions_punt1(columna_partial(element),1)).*k_vector))./(vect
or_posicions_punt2(fila,1)-vector_posicions_punt1(fila,1));

                    xa=xq+k_vector.*(xq-
vector_posicions_punt1(columna_partial(element),1));
ya=yq+k_vector.*(yq-
vector_posicions_punt1(columna_partial(element),2));

                    x_a_single=sqrt(((xa-
vector_posicions_punt1(columna_partial(element),1)).^2)+((ya-
vector_posicions_punt1(columna_partial(element),2)).^2));

vw_a_single=vector_velocitats(columna_partial(element)).*(1-
((D./(D+2.*k.*x_a_single)).^2).*(1-sqrt(1-Ct_lambda_optima)));

                    Dx=D+2.*k.*x_a_single;
                    r2=R;
                    r1=Dx./2;
                    d=sqrt((vector_posicions_punt1(fila,1)-
xa).^(2)+(vector_posicions_punt1(fila,2)-ya).^(2));
                    Awind=pi.*(r2.^2);
                    Ashad=(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)).*(r1.^2))+(acos(((r2.^2)+(d.^2)-
(r1.^2))./(2.*r2.*d)).*(r2.^2))-(sin(acos(((r1.^2)+(d.^2)-
(r2.^2))./(2.*r1.*d)))).*r1.*d;
                    betatjtk=Ashad./Awind;


vw_partial=vector_velocitats(columna_partial(element)).*(1-
sqrt(betatjtk.*(1-
vw_a_single./vector_velocitats(columna_partial(element))).^2));

                    sumes_arrel=sumes_arrel+((1-
vw_partial./vwfree).^2);
                end

                vector_velocitats(fila)=vwfree.*(1-
sqrt(sumes_arrel+((1-vw_single./vwfree).^2))); %MULTIPLE PARTIAL WAKE
EQUATION
```

ETSEIB

```
            end
        end
    end


    element_nan=0;
    cantitat_inicial_nan=size(find(isnan(vector_velocitats)),1);
    for element_nan=1:cantitat_inicial_nan
        array_nan=find(isnan(vector_velocitats));
        if element_nan<=cantitat_inicial_nan
            vector_velocitats(array_nan(element_nan))=0;
            cantitat_inicial_nan=cantitat_inicial_nan-1;
        end
    end
    %OK


    %cut out wind speed is 3m/s OK
    if size(find(vector_velocitats==0))==[0 1]
        for element_neg=1:nwt
            if vector_velocitats(element_neg)<3
                vector_velocitats(element_neg)=0.0000001;
            end
        end
    end



end

potencia_total=0;
for fila = 1:nwt
    if 0.5*Cp_lambda_optima*rho*A*(vector_velocitats(fila))^3 >
6000000 %Pnom
        potencia_total=potencia_total+6000000;
    else

potencia_total=potencia_total+(0.5.*Cp_lambda_optima.*rho.*A.*(vector_
velocitats(fila)).^3);
    end
end

%vector_velocitats

potencia_total;
%OK...

matriu_potencia=zeros(nwt,1);
for fila = 1:nwt
    if 0.5*Cp_lambda_optima*rho*A*(vector_velocitats(fila))^3 >
6000000 %Pnom
        matriu_potencia(fila)=6000000;
    else

matriu_potencia(fila)=(0.5.*Cp_lambda_optima.*rho.*A.*(vector_velocita
ts(fila)).^3);
    end
end
```

```
total_power=matriu_potencia;
%OK...
end
```

- **function_calling.m**

```
%calling the function
clear all;clc;
close('all')
format long

%% Normal calling

vwfree=9.5;
alpha=170;

fecamp_function(vwfree,alpha)

%% 1. vwfree vector

clear all;clc;format long;close('all');

alpha=0;
cutspeed=25;
vec_power=zeros([1 25]);


for element=1:cutspeed
    vwfree=element;
    vec_power(element)=fecamp_function(vwfree,alpha);
end

%%
plot(1:25,vec_power,  'g', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind speed [m/s]')
ylabel('Power [W]')
title(['Total power extracted in F√©camp with wind direction = 0
degrees'])
axis([0 25 0 600000000])


%% 2. alpha 360

clear all;clc;format long;close('all');

vwfree=9.5;
vec_power=zeros([1 73]);
itrange=360/5+1;

for element=1:itrange
```

ETSEIB

```
    alpha=5*element;
    vec_power(element)=fecamp_function(vwfree,alpha);
end
%%
plot(0:5:360,vec_power, 'g', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind direction [degrees]')
ylabel('Power [W]')
title(['Total power extracted in F√©camp (vwfree=9.5m/s)'])
axis([0 360 0 350000000])

%% 3. energy

v_vel_hub=[0.18 1.14 3.00 5.33 8.08 11.25 14.78 18.61 21.61];
v_ang_hub=[0 22.5 45 67.5 90 112.5 135 157.5 180 202.5 225 247.5 270
292.5 315 337.5];
matriu_power=zeros([16 9]);

for element=1:length(v_vel_hub)
    for element2=1:length(v_ang_hub)

matriu_power(element2,element)=fecamp_function(v_vel_hub(element),v_an
g_hub(element2));
    end
end

%for element=1:length(v_vel_hub)
    %matriu_power(1,element)=fecamp_function(v_vel_hub(element),0);
%end

%%

v_ang_hub=[0 22.5 45 67.5 90 112.5 135 157.5 180 202.5 225 247.5 270
292.5 315 337.5];
energy_sum=[499833985000000 508861378000000 494285380000000
529217968000000 283437731000000 129813592000000 255251883000000
241919410000000 961597643000000 652112596000000 1399552232000000
1782070969000000 1229897461000000 377906734000000 286733786000000
123743489200000];
energy_sum=energy_sum./1000000;

plot(v_ang_hub,energy_sum,  'r', 'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Wind direction [degrees]')
ylabel('Energy [MW¬∑h]')
title(['Total energy extracted in F√©camp during 1 year'])
axis([0 360 0 2000000000])

%% surf energy

aba=readtable('/Users/davidbigorda/Desktop/TFG/Matlab_TFG/power_data.c
sv'); %filas cada angulo (16 intervalos) i columnas cada velocidad (9
int)
aba2=table2array(aba);
```

```
surf(aba2)
axis on
xlabel('Wind speed [m/s]')
ylabel('Wind direction [degrees]')
zlabel('Energy [W¬∑h]')
title(['Total energy extracted in F√©camp during 1 year'])
axis([0 9 0 16 0 600000000])


%% 4. lambda (Cp i Ct)

alpha_rango=[0 30 60 90 120 150 180 210 240 270 300 330];
lambda_rango=2:0.5:9;
vwfree=9.5;
vector_power_lambda=zeros([1 length(lambda_rango)]);


for element=1:length(lambda_rango)
    for element2=1:length(alpha_rango)

vector_power_lambda(element2,element)=fecamp_function_cp(vwfree,alpha_
rango(element2),lambda_rango(element));
    end
end

%%

vector_power_lambda_total=[363451000000 1853040000000 4944120000000
9514320000000 14934800000000 20445900000000 25413000000000
24690200000000 30260097000000 31999200000000 32410600000000
31658888000000 29481745000000 29471000000000 23284100000000];
lambda_rango=2:0.5:9;

plot(lambda_rango,vector_power_lambda_total/1000000,  'r',
'LineWidth', 2)
axis on
%axis equal
%grid off
%text(0:5:360, vec_power, vec_power(:,1), 'Fontsize', 8);
xlabel('Lambda value')
ylabel('Power [MW]')
title(['Power generated in F√©camp with vwfree=9.5 m/s'])
axis([2 9 0 40000000])


%% bubbles energia x turbina

v_ang_hub=[0 22.5 45 67.5 90 112.5 135 157.5 180 202.5 225 247.5 270
292.5 315 337.5];
v_vel_hub=[0.18 1.14 3.00 5.33 8.08 11.25 14.78 18.61 21.61];

matriu_hores=[2 30 105 125 100 48 17 3 0;1 26 97 173 208 107 25 3 1;0
21 90 187 178 77 18 3 1;2 28 108 163 115 45 12 3 0;0 18 83 112 55 21 5
0 0;1 24 86 106 58 14 3 0 0;0 21 83 117 68 25 6 1 0;0 18 78 124 106 48
19 3 1;4 30 87 143 161 117 51 13 2;1 21 79 142 205 174 85 24 6;2 25
104 216 273 216 108 31 5;0 21 111 253 331 211 94 30 7;2 34 131 228 246
121 58 17 3;1 25 98 128 103 54 23 7 2;1 29 98 101 73 41 19 5 1;0 19 82
89 66 37 15 2 0];
```

```
nwt=83;
matriu_energies=zeros(nwt,1);



for element=1:length(v_vel_hub)
    for element2=1:length(v_ang_hub)

matriu_energies=matriu_energies+(fecamp_function(v_vel_hub(element),v_
ang_hub(element2)).*matriu_hores(element2,element));
    end
end

%xlswrite(filename,matriu_energies,1,'A1')
```