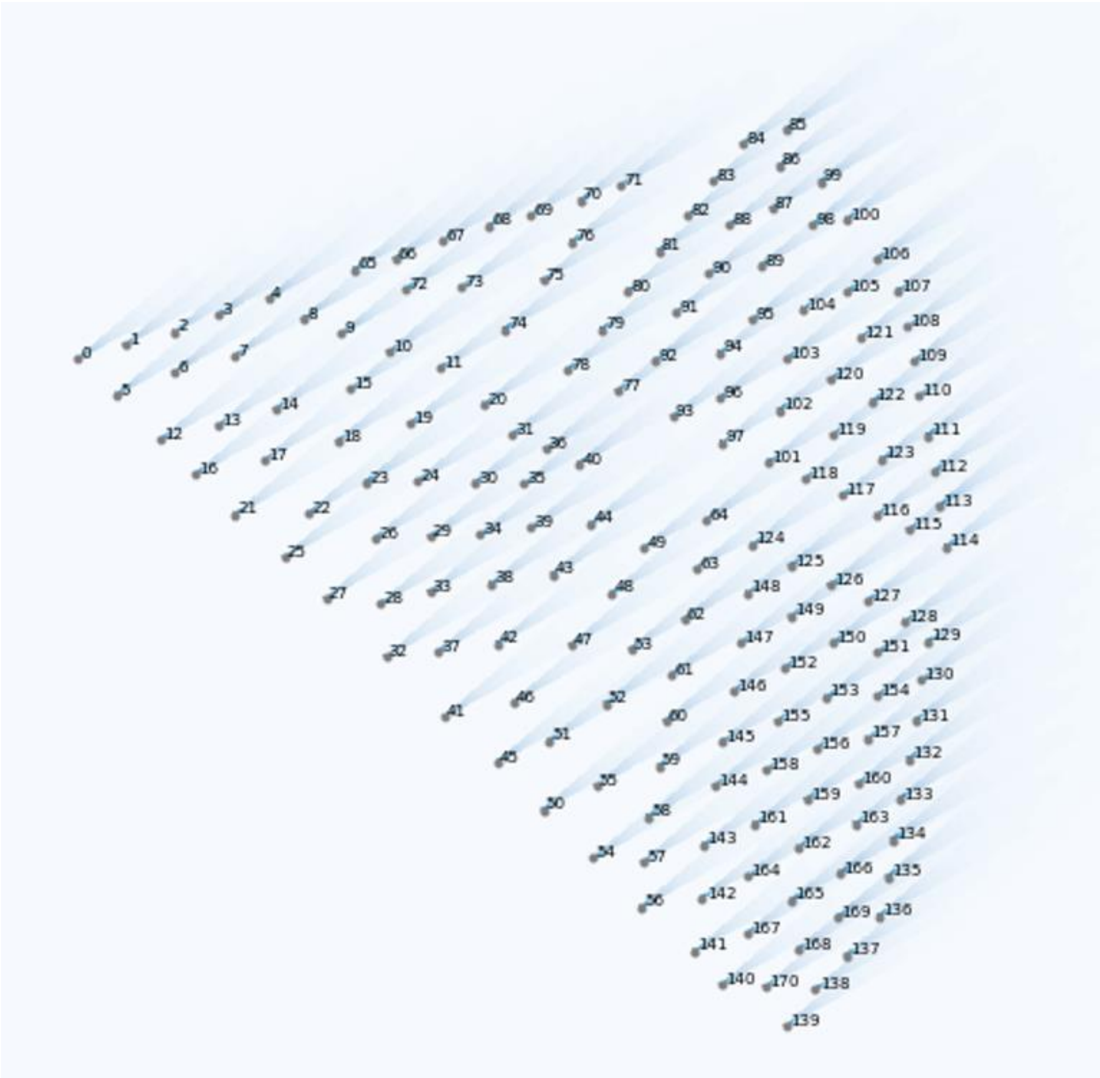
FMH606 Master's Thesis 2024

Process Technology

Offshore Wind Farms Modelled for Simulations with PyWake



Stian Juvet Sørensen

Faculty of Technology, Natural sciences and Maritime Sciences

Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2024

Title: Offshore Wind Farms Modelled for Simulations with PyWake

Number of pages: 80 (54 + Appendices)

Keywords: Offshore wind farms, PyWake, wind turbine

Student: **Stian Juvet Sørensen**

Supervisor: **Geir Elseth, Equinor**

External partner: **Atle Gyllensten, Equinor**

Summary:

With the shift to carbon neutral energy global wind electricity production is rapidly increasing. Leading to larger and larger wind turbines and wind farms. As wind conditions change it is important to understand how production at wind farms will be affected by the local wind conditions, to optimize the layout and production of the wind farm. PyWake is a tool that can be used to simulate flow fields and energy production at wind farms.

This thesis aims to gather data and information about offshore wind farms, then use the information gathered to make models capable of running simulations with PyWake. Results from the simulations will be compared with the production of the actual wind farms.

To estimate the annual energy production (AEP) the NOJ wake deficit model have been used. To make the layout of the wind farms being studied, the 4COffshore global map have been used. Publicly accessible data have been found and used to make models of the turbines at the different wind farms. And Global Wind Atlas have been used to gather information about the wind conditions needed to make the models.

The offshore wind farms Borssele I & II, Borssele III & IV, the Borssele wind farm zone, Borkum Riffgrund II and Hornsea Project 2 have been modelled in this thesis. The NOJ wake deficit model have been used to estimate the AEP, and that have been compared to information about the actual production at the different wind farms. As in previous studies the NOJ model estimates a higher AEP than the actual AEP. The models are also shown to be compatible with different PyWake functions.

Preface

This master thesis was written during the spring of 2024 as the final project to complete a MSc. In Process technology at the University of South-Eastern Norway (USN).

I would like to thank my supervisors professor Geir Elseth and external supervisor Atle Johnsen Gyllensteen from Equinor for their help, guidance and insights during this project.

Porsgrunn, 14.05.2024

Stian Juvet Sørensen

Contents

List of figures

List of tables

1 Introduction

As the world looks to reduce carbon emissions and increase electricity production from renewable sources, offshore wind provides an attractive alternative. Offshore wind is a rapidly expanding market with the possibility of meeting the electricity demand of today many times over[1]. With flatter terrain and higher wind speeds, wind farms located at sea have preferable conditions compared to their onshore counterparts. A total of 25 new offshore wind farms with a combined capacity of 9.8 GW became operational in 2023[2]. This took the total number up to 282 operational offshore wind farms and the global capacity of offshore wind farms up to 67.4 GW as seen in Figure 1-1. ERM predicts the total capacity of operational offshore wind farms worldwide will reach 250 GW in 2030 and 2 465 GW by 2050[3].

As the technology continues to advance and wind turbines gets larger and more powerful, offshore wind becomes more cost competitive. In 2022 the average size of new offshore wind farm was 225 MW, this increased to 392 MW in 2023.

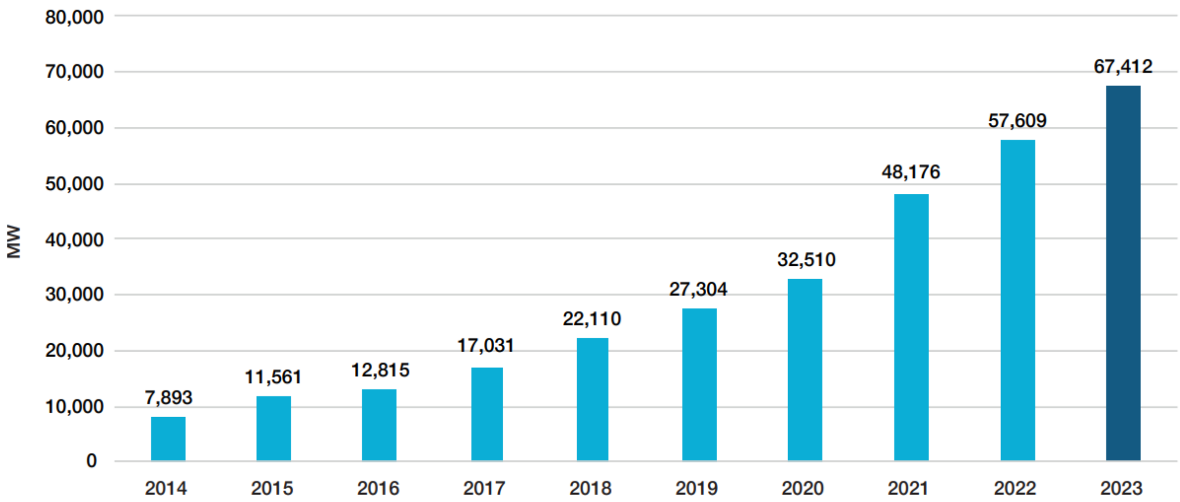


Figure 1-1 show the combined capacity of installed offshore wind farms in operation[2].

Wind and offshore wind continues to grow and make up a larger part of the electricity mix, producing 7.33% of worldwide electricity generation in 2022 and 7.82% in 2023[4, 5]. This is a number that is expected to continue rising as countries continue to invest into carbon neutral electricity production. Offshore wind have the same weaknesses as onshore wind, that electricity production varies due to wind speeds, wind direction and reduction in production due to wakes of other turbines. Therefore it is important to understand how a wind farm will perform, and how to optimize design of wind farms to maximize production and reduce losses. One tool that can help with these predictions is PyWake, PyWake allows for simulations of exiting, proposed or theoretical wind farms. The purpose of this thesis is to gather information and use this to make models of existing offshore wind farms that are compatible and can be used with PyWake.

1.1 Task description

The task description for this thesis is dived into three parts:
Part 1: Literature study.

- Collect offshore wind farm production data from public sources and publications. Any data is of interest, like energy production, power production, wind measurements etc.
- Get a best possible description of the farms in question, including turbine positions, WTG type etc.

Part 2: Create and publish a database.

- Collect all data in an easily accessible database.
- Create a Python tool to access the data.
- The tool should provide the input needed for running PyWake simulations in an easy way.
- Publish data and the Python tool on GitHub.

Part 3 (if time allows it): Compare simulation results to production.

- Use PyWake to run wake simulations for comparing simulated production to the actual production.

To make the data as easily accessible and user friendly as possible, the data and information will be used to make models of the offshore wind farms described in the thesis. The wind farms modelled in this thesis will be modelled in the same way as already exiting wind farms in PyWake. This makes them compatible with PyWake without the need for an additional tool, and can be used in the same way as wind farms shown in the existing PyWake tutorial making them as easily accessible and user friendly as possible.

2 Theory

In this chapter the information and theory needed to model the offshore wind farms will be presented.

2.1 Wind turbines

In this chapter the different wind turbines modelled will be described and the properties needed to make models of them will be presented.

2.1.1 Vestas V164-8 MW

Vestas V164-8 MW is a wind turbine made by Vestas. It is a pitch-controlled turbine, which means the angle of the blades adjusts to keep the rotational speed of the wind turbine within the operating limits[6]. The rated power of the wind turbine is 8000 kW. Cut in speed which is the wind speed where a wind turbine begins producing electricity, for the V164-8 MW turbine it is 3,5 m/s. Cut off speed is the wind speed where a wind turbine will stop producing to prevent electrical and mechanical damage is 25 m/s for this turbine. The rated speed is the wind speed needed for a turbine to produce at its maximum capacity and is 13 m/s for the V164-8 MW wind turbine. Figure 2-1 shows the power curve of the V164-8 MW turbine. The diameter of the V164-8 MW is 164 meters[7]. Height of the hub on the turbine is site specific and varies from 105 to 140 meters[8]. As of the 30. Of September 2022 there have been installed 275 Vestas V164-8 MW wind turbines globally[9]. This number includes turbines off and onshore turbines. As the thrust coefficient curve of the turbine isn't publicly available, the thrust coefficient curve of the LW 8 MW reference turbine will be used[10]. This is a turbine model made based on published data of the Vestas V164-8 MW wind turbine and have been validated DNV-GL which is an independent provider of wind turbine engineering services. The values of the thrust coefficient curve are shown in Table 2-1.

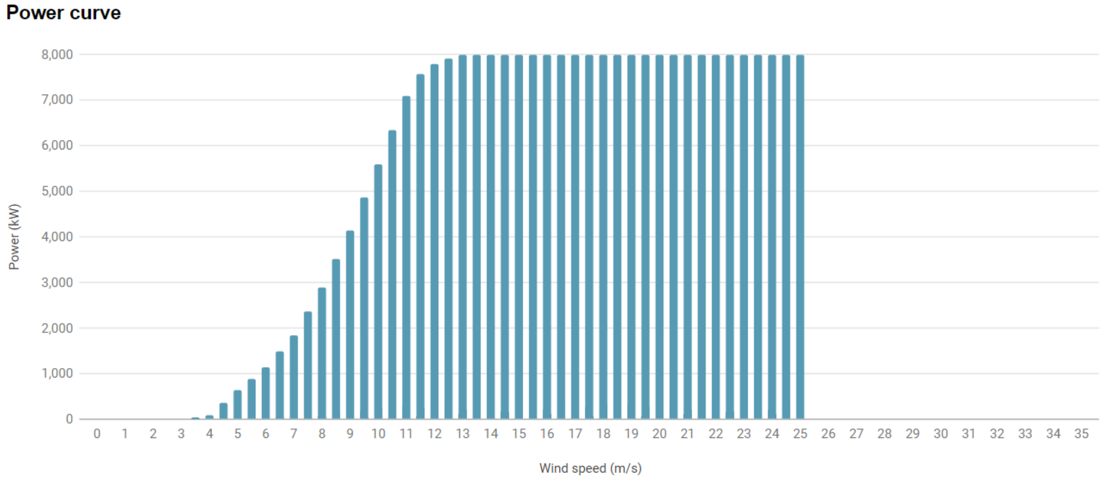


Figure 2-1 shows the power cur of the Vestas V164-8 MW wind turbine[7].

Table 2-1 show the values of the thrust coefficient curve of the LW 8 MW reference wind turbine[10].

Wind Speed[m/s]	Ct
4	0.92
5	0.85
6	0.82
7	0.80
8	0.78
9	0.76
10	0.73
11	0.67
12	0.52
13	0.39
14	0.30
15	0.24
16	0.19
17	0.16
18	0.14
19	0.12
20	0.1
21	0.09
22	0.08
23	0.07
24	0.06
25	0.05

2.1.2 Vestas V164-9.5 MW

Vestas V164-9.5 MW is a wind turbine produced by Vestas, it is a continuation and improvement of the previous V164 models[11]. Cut in wind speed for the turbine is 3.5 m/s and the cut off wind speed is 25 m/s while the rated wind speed is 14 m/s. Vestas V164-9.5 MW is a pitch-controlled turbine have a rated power of 9500 kW. The rotor of the wind turbine has a diameter of 164 meters. Figure 2-2 shows the power curve of the of the turbine[12]. The hub height varies and is site specific, heights varies from 105 meters and upwards to 140m. As of September of 2022 there have been installed 297 Vestas V164-9.5 MW wind turbines, both offshore and onshore[9]. Some of the offshore wind farms where the turbine have been installed include Northwester 2 in Belgium, Borssele III, IV and V in the Netherlands, Kincardine Offshore Windfarm, Moray East and Triton Knoll Wind Farm in the United Kingdom. Hai Long 2A, 2B and 3 in Taiwan have also installed V164-9.5 MW turbines.

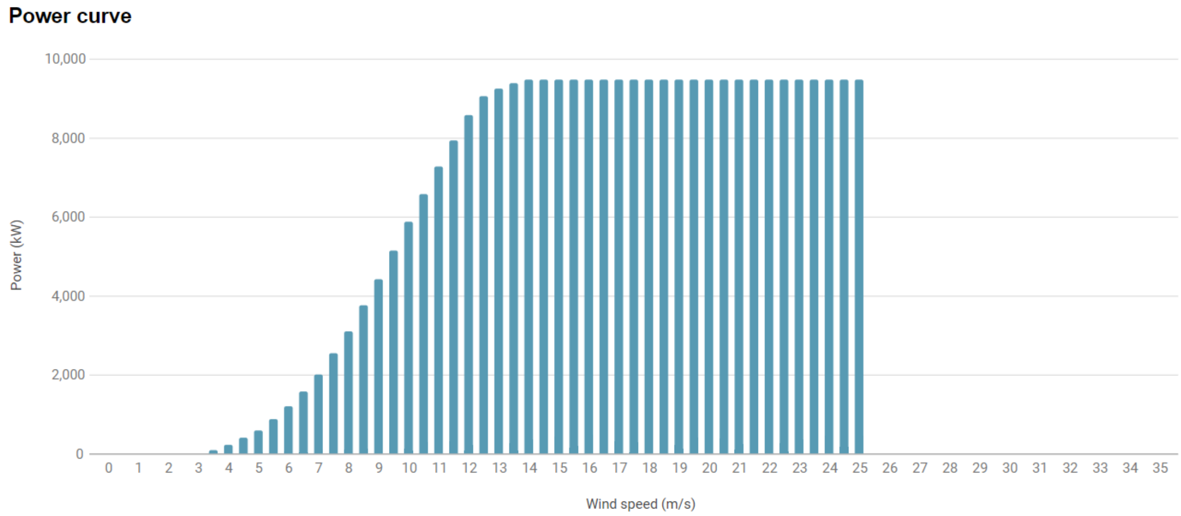


Figure 2-2 shows the power curve for the Vestas V164-9.5 MW wind turbine[12].

The thrust coefficient curve for the V164-9.5 MW turbine is not publicly available so a curve from a comparable wind turbine will be used in the model. The turbine chosen is a reference turbine made by the Danish Technical University already implemented in PyWake called DTU10MW[13]. This was chosen due to its similar size to the V164-9.5 MW turbine and as its already implemented and used in PyWake.

2.1.3 SG 8.0-167 DD

The SG 8.0-167 DD is an offshore wind turbine made by Siemens Gamesa. The rotor of the turbine have a diameter of 167 meters and rated power of the turbine is 8000 kW[14]. Sg 8.0-167 DD is pitch controlled and starts producing electricity when the wind speed reaches 3 m/s, maximum electricity production or rated speed is reached at 12 m/s wind speed. Figure 2-3 shows the power curve, it shows that the cut off speed is 25 m/s, Siemens state that the SG 8.0-167 DD utilizes the High Wind Ride Through (HWRT) system, which is a system that ramps down production when wind speeds are higher than 25 m/s. This is done by pitching the blades and limiting the rotational speeds of the turbine[15, 16]. The HWRT changes the power curve as shown in Figure 2-4 but a power curve with the HWRT system is not publicly available. Therefore, the power curve shown in Figure 2-3 will be used in the modelling of the wind turbine. The hub height of the turbine is site specific according to Siemens Gamesa, at the Østerild wind test cite there have been tests with a SG 8.0-167 DD with a hub height of 120 meters.

Sites like Borssele 1 and2 in the Netherlands, Kaskasi in Germany, Hywind Tampen in Norway, Hornsea Project two in the United-Kingdom and Sunrise wind in the united states all have installed the SG 8.0-167 DD wind turbine[14].

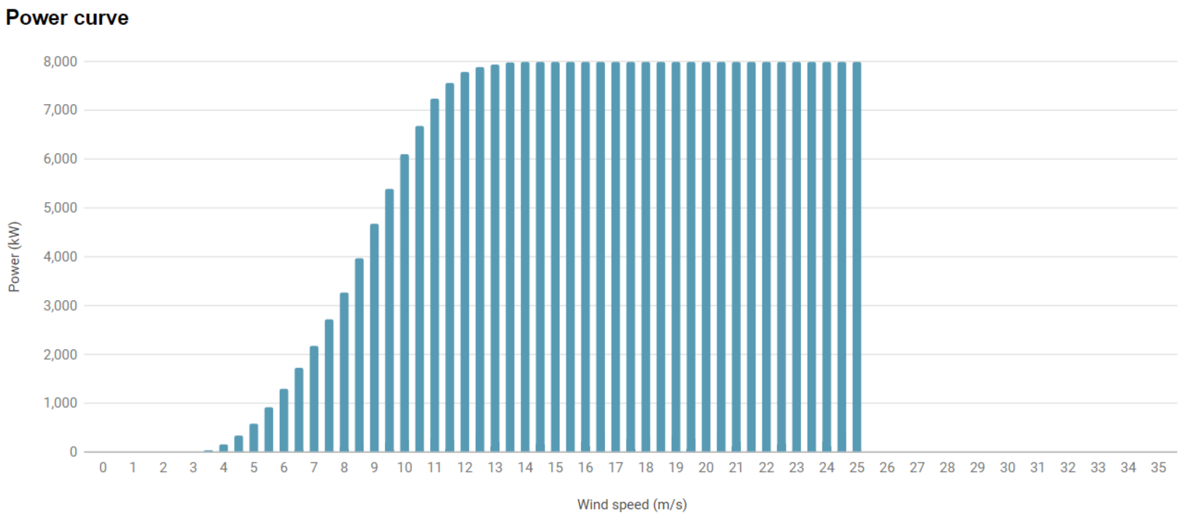


Figure 2-3 shows the power curve of the SG 8.0-167 DD wind turbine[14].

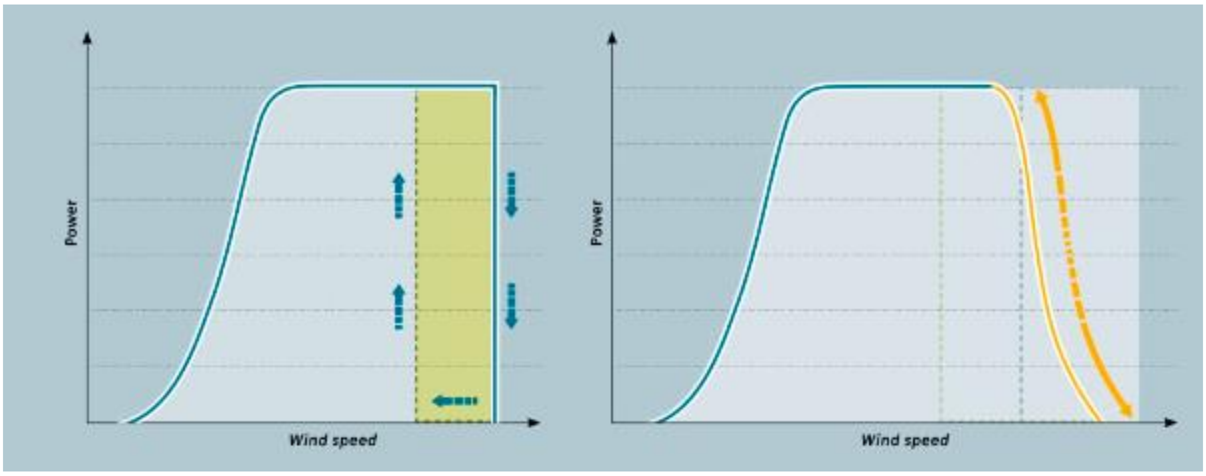


Figure 2-4 shows power curve without and with the high wind ride through system[15].

2.2 Windfarms

This chapter will provide an overview of the windfarms that have been studied and modelled to be compatible with PyWake. The windfarms have been chosen based on criteria discussed with the supervisors. Criteria for selection of the windfarms were that newer windfarms with larger turbines were preferred, and wind farms with higher amounts of turbines was preferred compared to farms wind fewer turbines. Windfarms where production data available was also favoured due to the possibility of comparing the model's estimated production with the real electricity production.

2.2.1 Borssele Offshore Wind Farm III & IV

Borssele III and IV is an offshore wind farm located in the Borssele wind farm zone 22 km off the coast of Zeeland in the Netherlands[17, 18]. The zone consists of five different sites with three different wind farms, the layout of the zone and sites are shown in Figure 2-5. Construction of Borssele III and IV started in 2019, and the last turbine was installed at the end of 2020. The wind farm became fully operational in February of 2021[19]. The wind farm consists of 77 MHI Vestas 164 9.5 MW wind turbines mounted on monopiles. Total capacity of the Borssele III and IV is 731,5 MW. It was developed and is operated by the Blauwind consortium and is stated to produce 3 TWH annually, which is estimated to power 825 000 households. The hub height of the turbines at Borssele III and IV is 109 meters[20]. The turbines are built in depths varying from 16 to 38 meters.

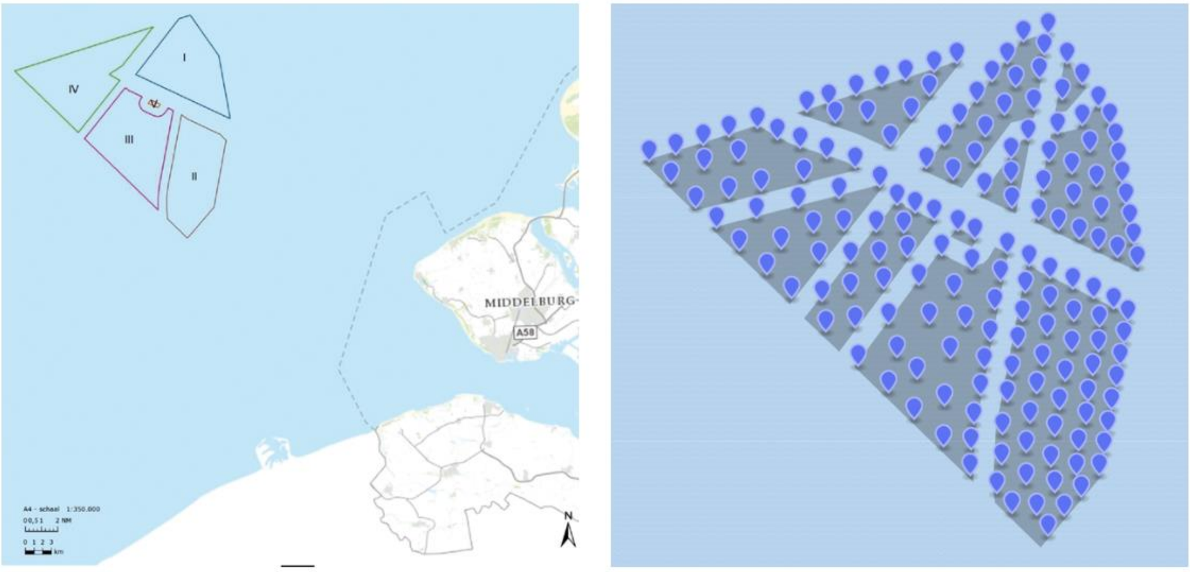


Figure 2-5 shows the different sites at the Borssele wind farm zone[21, 22].

2.2.2 Borssele I and II

Borssele I and II is a wind farm with a total capacity of 752 MW, located in the Borssele wind farm zone 55 km of the coast of Zealand in the Netherlands. The wind farm consists of 94 SG 8.0-167 DD wind turbines. When it was commissioned in November 2020 it was the largest offshore wind farm in the Netherlands and the second largest in the world[23]. It was the first offshore wind farm operated by Ørsted in the Nederland's. Ørsted states it will produce enough electricity to power one million Dutch homes. As the average Dutch household uses approximately 3 000 kWh of electricity in a year[24], this means the Borssele I and II wind farm produces around 3 000 GWh of electricity a year.

The turbines at the wind farm have a tip height of 200 meters and an hub height of 116.5 meters[25]. All 94 turbines are mounted on monopile foundations at water depths ranging from 14 to 36 meters. The layout of the wind farm is shown in Figure 2-5, where it can be seen that that the wind farm is taking up two zones in the Borssele wind farm zone. With Borssele III and IV taking up another two zones and the last zone, Borssele V being a zone with just two wind turbines designated to be an innovation site[26].

2.2.3 Borkum Riffgrund II

Borkum Riffgrund II is an offshore wind farm located 54 km offshore of north Germany in the north sea[27]. The wind farm consists of 56 V164-8 MW MHI Vestas wind turbines with a hub height of 105 meters. The wind farm has total capacity of 450 MW. It was completed and commissioned at the end of 2018, and is owned and operated by Ørsted with partners Gulf energy and Keppel infrastructure trust[28]. The location of Borkum Riffgrund II and layout for the turbines is shown in Figure 2-6. The turbine layout is found by using the global offshore map from 4coffshore by TGS[29]. Figure 2-6 shows that 52 of the 56 wind turbines are located to the west and south of the Borkum Riffgrund I wind farm, while the remaining four turbines are to the north-east of the Borkum Riffgrund I wind farm.

Table 2-1 show the annual electricity produced at the Borkum Riffgrund II wind farm from 2019 to 2023[30].

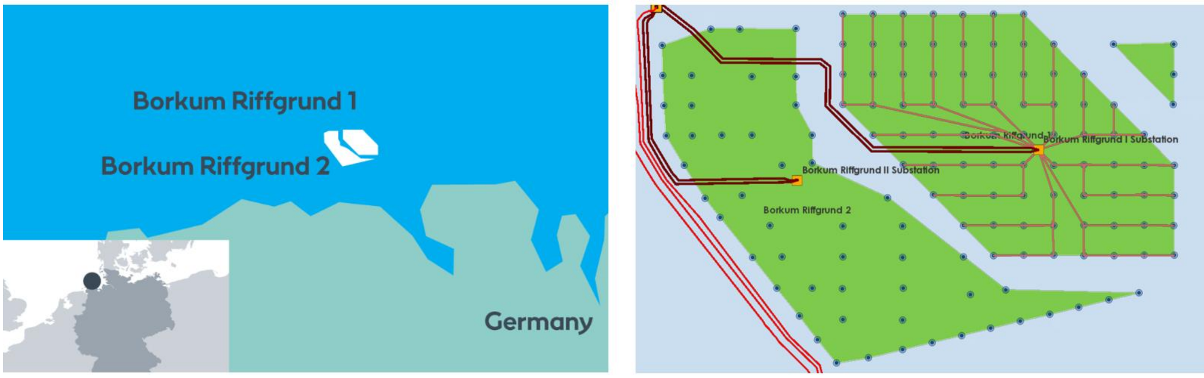


Figure 2-6 shows the location of the Borkum Riffgrund II wind farm[28], and the layout of the wind turbines from the 4COffshore global map[29].

Table 2-2 show electricity production at the Borkum Riffgrund II by year[30].

Year	**Electricity produced in TWh**
2019	**1.165**
2020	**1.450**
2021	**1.297**
2022	**1.286**
2023	**1.345**

2.2.4 Hornsea Project 2

Hornsea project 2 is an offshore wind far located in the Hornsea Zone almost 90 km of the coast of England in the North Sea, as seen in Figure 2-7[31]. Ørsted is the developer and operator of the two current wind farms in the zone, with a third project under construction and a fourth in the planning stages. Construction of Hornsea 2 started in 2020 and the wind farm became fully operational in 2022. With a total capacity of 1.32 GW it is the largest offshore wind farm in the world. The wind farm consists of 165 Siemens Gamesa 8.0-167 DD wind turbines arranged as seen in Figure 2-8. Water depths in the Hornsea Zone ranges mostly between 30 and 40 meters but reaches depths of 70 meters. The tip height of the turbines at the site reaches 200 meters and the hub height is 116 meters[32].

In 2023 Hornsea 2 produced 12.2% of the total 49 TWH electricity produced from offshore wind in the UK, according to the UK offshore wind report 2023[33]. This means the electricity production from Hornsea 2 in 2023 was 5 978 GWh, which is enough to power more than 1.4 million homes in the UK according to Ørsted[31].

Figure 2-7 show the location of the Hornsea Zone[31].

Figure 2-8 show the wind turbine layout of Hornsea project 1 and 2 from the 4COffshore map[29].

2.3 Wake effects

To produce electricity wind turbines converts energy from wind, this leaves the wind after the turbine with a lower energy content than the wind upwind of a turbine. This leads to the wake from turbines being more turbulent and having lower wind speeds. When the wind keeps moving further away from a turbine it gradually returns to the conditions it was before the turbine, or free stream conditions. The drop in wind speed and energy causes loss in power production for turbines located in the wake downwind of other turbines. The increased turbulence can also cause additional mechanical load on downwind turbines. Therefore, it is important to study and understand the wake effects when designing windfarms to maximize electricity production and the lifetime of the wind turbines. Figure 2-9 shows the airflow around an idealized turbine, where the area behind the turbine is the wake. The wake from multiple turbines can also combine as shown in Figure 2-10, this is called wake merging[34]. To the left in Figure 2-10 full wake merging is shown. This happens when the wind direction is parallel with turbines in a row. In the middle wake-turbine merging is shown and to the right wake-wake merging is shown. These are wake scenarios that can happen and affect electricity production and vary as the wind direction at the wind farm changes. In wind farms where the turbines are organized in a grid, wind directions causing full wake merging leads to a steep reduction in electricity production[35].

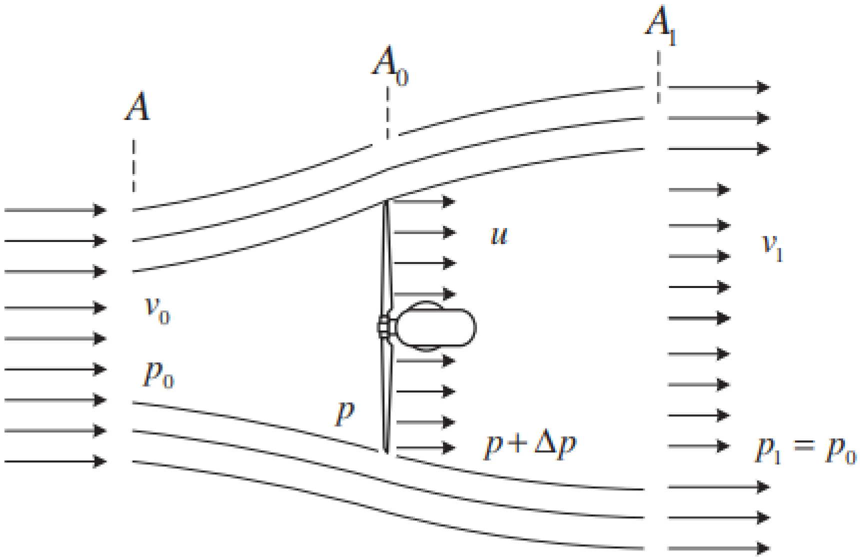


Figure 2-9 shows air flow around an idealized turbine with pressure and velocity[35].

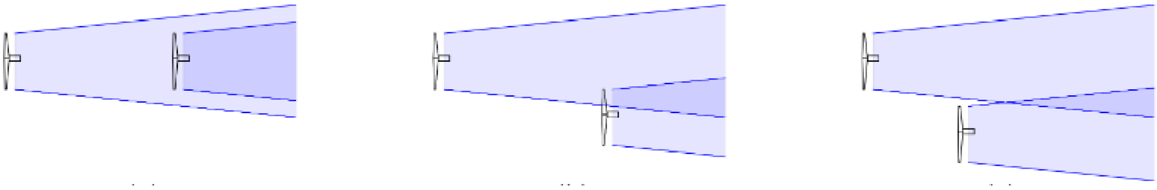


Figure 2-10 shows wake from different turbines merging[34].

The wake after a wind turbine can be separated into two different regions, the near wake and the far wake with a transition zone between them. This is shown in Figure 2-11. The near wake is a complex region where the wind speed and pressure are low. From 1 to 2 rotor diameters after the turbine the turbulence in the boundary layer gets mixed with the higher speeds of the free stream wind. This creates shear layer that moves as seen in Figure 2-11[36]. When the shear layer meets in the centreline behind the turbine, from 2 to 5 rotor diameters after the turbine it marks the end of the near wake. In the transition zone the velocity of the wake continues to increase. After the transition zone the wake enters the far wake, at more than five rotor diameters from the turbine the far wake region begins. The wake has finished developing and the wind speed difference is gradually reduced until it reaches free stream velocity again. It is this region that affects other wind turbines and windfarms in proximity. The wake from wind farms usually extends up to 15 km down wind of the wind farm[37], in the study "Using Satelite SAR to Characterize the Wind Flow around Offshore Wind farms" it was found wakes extending up to 55 km[38].

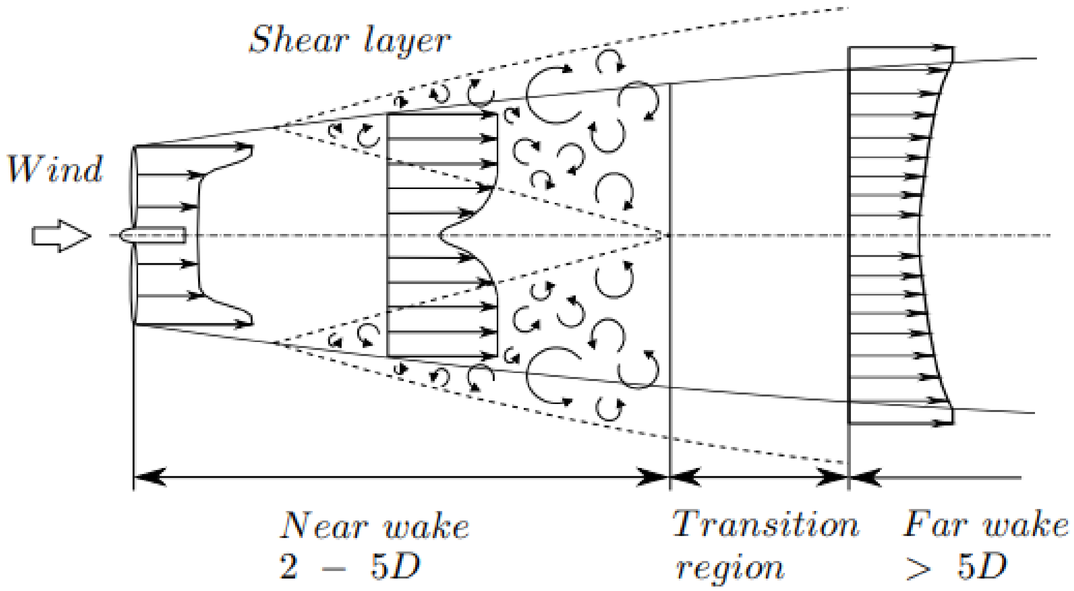


Figure 2-11 show the different wake regions after a wind turbine[36].

2.4 PyWake

PyWake is a tool developed by Danmarks Tekniske Universitet (DTU) which uses Python to simulate windfarms. It is an open-source tool that can be used to simulate flow fields in windfarms, estimate annual energy production of wind farms, optimize the layout of wind farms, estimate the noise made by wind turbines and farms and estimate the electricity production of individual wind turbines[39, 40].

2.4.1 PyWake philosophy

PyWake is made to be modular as seen in Figure 2-12, to give users flexibility and options to choose different parameters or cases to be studied. The Windfarmmodel is the central part of PyWake`s architecture. Windfarmmodel takes information from Site and WindTurbines objects and returns a SimulationResult object.

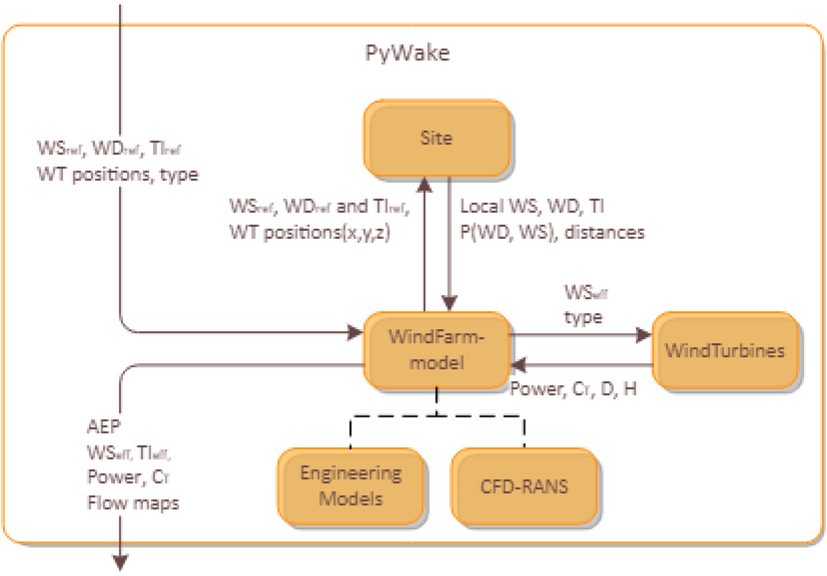


Figure 2-12 architecture of PyWake[40].

The site object contains information about the number of turbines, the position of the turbines, turbulence intensity. Information of the probability of the different wind directions and wind speed combinations are used to provide local wind speeds and wind directions. The Site object calculates down wind, cross wind and vertical distance between the wind turbines in the windfarm being studied. PyWake allows for different ways to specify the wind conditions at sites of wind farms with different complexities. The models made in this thesis will be UniformWeibullSite`s. These are sites where the wind speed is sector dependent and the wind speed is Weibull distributed[41, 42]. One of the predefined sites in PyWake is the Hornsrev1 offshore wind farm. This is modelled as a uniform Weibull site where the wind speeds are Weibull distributed, the wind direction is defined with the probability of each sector. The Weibull distribution is often used to describe the variations over long periods of time in wind speed. The Weibull distribution formula uses two parameters A and k and is shown in formula (2.1)[43].

$$f(v) = \frac{k}{A}\left(\frac{u}{A}\right)^{k-1} \exp\left(-\left(\frac{u}{A}\right)^{k}\right) \tag{2.1}$$

Where u is the wind speed, A is the scale parameter and k is the shape parameter. Weibull scale parameter is a measure of the characteristic wind speed for the wind speed distribution at a site. The Weibull shape or form parameter, k, describes the shape of the Weibull distribution curve. Constant wind speeds will give a high k number while if the wind speeds vary the k number will be lower[44-46]. Shape and scale parameter can be calculated with formula (2.2) and (2.3)[47].

$$k = \left[\frac{\sum_{i-1}^{n} u_i^k \ln(u_i)}{\sum_{i-1}^{n} u_i^k} - \frac{\sum_{i-1}^{n} \ln(u_i)}{n}\right]^{-1} \quad (2.2)$$

$$A = \left(\frac{1}{n}\sum_{i-1}^{n} u_i^k \right)^{\frac{1}{k}} \quad (2.3)$$

Where u_i is the wind speed at the time step and n is the number of wind speed measurements.

The WindTurbines object takes information about the type of wind turbine and returns the height of the hub and diameter of the turbine, power curve and thrust coefficient curve.

In PyWake the engineering models consists of either one or two wind farm models combined with a wake deficit model, it can also have a turbulence model, superposition model, blockage deficit model, rotor average model, deflection model or ground model. These models are optional while the wake deficit model and one of the Engineering Windfarm models (PropagateDownwind or All2AllIterative) are necessary. All models are described in the PyWake tutorial section of the PyWake site. This makes it possible for many combinations of models to be studied and how they affect simulation results. The different engineering models are shown in Figure 2-13[40].

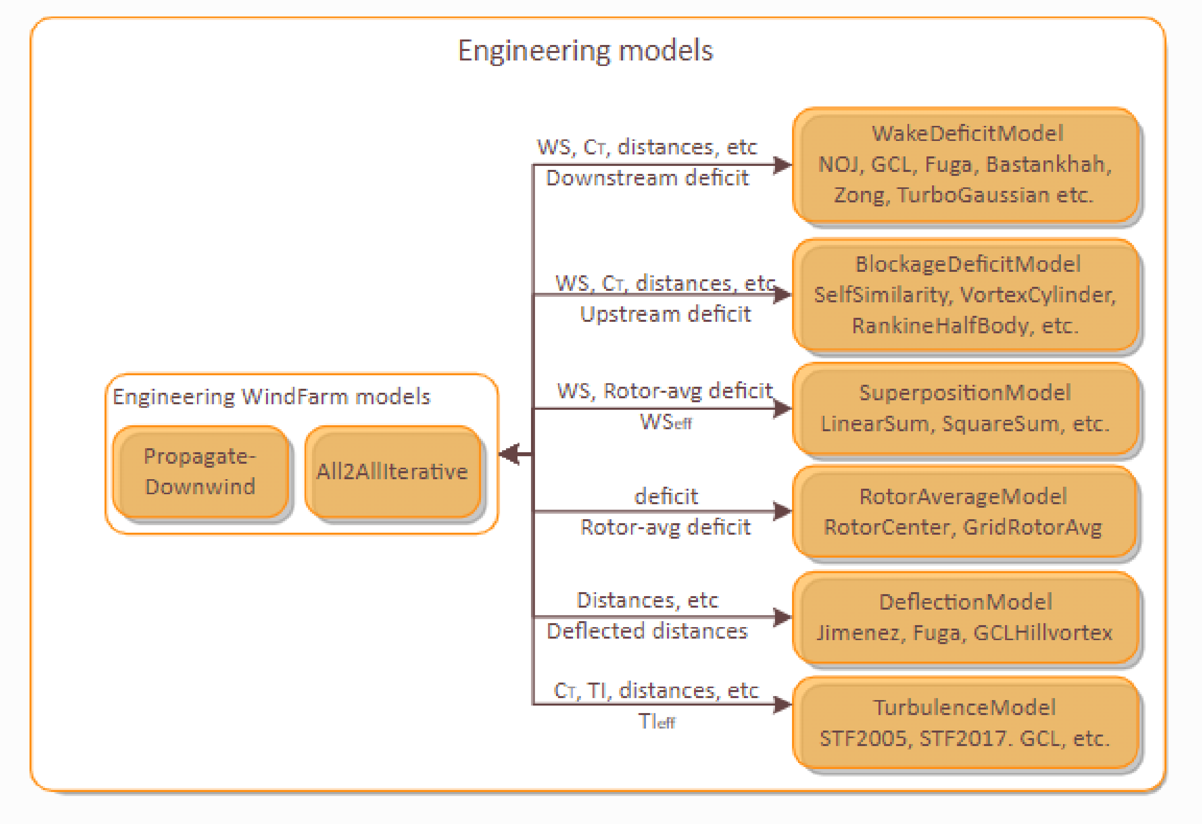


Figure 2-13 shows the different engineering models in PyWake[40].

The simulation result object that provides information about the simulation in an xarray dataset. Some of the information it provides is wind speed and direction studied, wind turbine numbers and AEP of each turbine and the AEP of all the wind turbines combined. The simulationResult() and aep command also makes it possible to calculate the AEP for the wind farm if wake loses are ignored. This makes it possible to find the wake losses by subtracting the AEP with wake losses from the calculation ignoring the losses[48].

2.4.2 Wake deficit models

Wake deficit models are used to compute the wake effects. The reduced wind speeds cause reduced power production in turbines that are located down wind. Therefore, it is important to study these effects when designing a wind farm to optimize power production and make the layout in way that minimizes the wake losses for the turbines.

PyWake offers multiple different wake deficit models to choose from. One of the most common models to use is the Jensen model, which is called NOJ in PyWake. This is a model made from “A note on wind generator interaction.” by Niels Otto Jensen in 1983. This is a very common model used due to its simplicity and being well documented. One of the simplifications the Jensen model does is to neglect the immediate wake behind the turbine, the near wake, this allows for the wake to be treated as turbulent and makes the model only valid in the far wake. The Jensen model assumes that the wake linearly expands in diameter[49, 50]. Figure 2-14 shows the Jensen wake model to the right compared to the flow around an idealized wind turbine on the left[34].

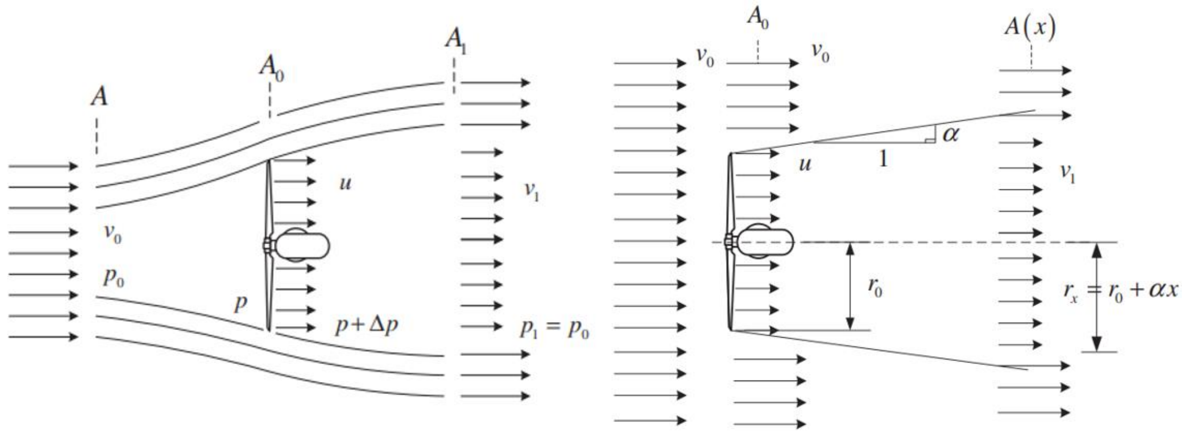


Figure 2-14 shows how air flow around an idealized wind turbine compared to the wake of the Jensen model[35].

The velocity of the wake can be found according to the Jensen model by equation (2.4). Where U is the free stream velocity, r_0 is the radius of the wake, α is the entrainment constant and x is the distance downwind from the wind turbine[50, 51].

$$V = U\left(1 - \frac{2}{3}\left(\frac{r_0}{r_0 + \alpha x}\right)^2\right) \quad (2.4)$$

In the master thesis by Mersal Al Halabi titled “*Comparing wind farm production data to engineering wake model simulations* “ from 2023[52], it was found that when using the same parameters the NOJ wake model estimated an higher annual energy production (AEP) of a wind farm compared to other wake models in PyWake. The models being compared was NOJ, BastankhahGaussian and the TurbOPark wake models. All of the models being investigated estimated higher AEP than the actual AEP of the wind farm being simulated. The NOJ model estimated a 23.89% higher AEP of the Horns Rev 1 windfarm than the actual AEP in the study.

2.5 Global Wind Atlas

When modelling a wind farm, it is important to have information about the wind conditions at the site of the wind farm. To gather information about wind conditions at the different sites the web-based Global Wind Atlas (GWA) app can be used[53, 54]. GWA is made and owned by the Technical University of Denmark and uses data from Vortex. It is released with the World Bank Group and funded by the Energy Sector Management Assistance program[54, 55]. From the Global Wind atlas site, it is possible to gather different information and data about chosen locations at different heights (10, 50, 100, 150 and 200 meters).

To model wind conditions in PyWake the wind sector frequency at the wind farm is needed, this can be found at the GWA when choosing the location of the wind farm. After choosing the location the wind frequency rose can be examined. Figure 2-15 shows an example of a wind frequency rose of one location, the rose shows the probability of wind from the different directional sectors.

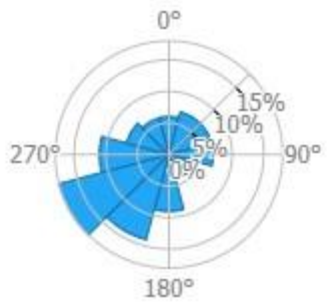


Figure 2-15 shows wind frequency rose from GWA at a chosen location[55].

Other necessary information about wind conditions needed for a wind farm in PyWake is the Weibull scale parameter, A and the Weibull shape parameter, K. This information is also available at the Global Wind Atlas site, by downloading a generalized wind climate (GWC) file for a chosen location. The GWC file gives access to data derived from the closest mesoscale grid cell to the chosen location. From the file the Weibull A and K parameters can be found for each wind direction sector, for different heights and roughness's[56]. Figure

2-16 shows how the vertical wind speed profile is affected by different roughness's. It is also shown in Figure 2-16 examples of the roughness of different ground areas[57].

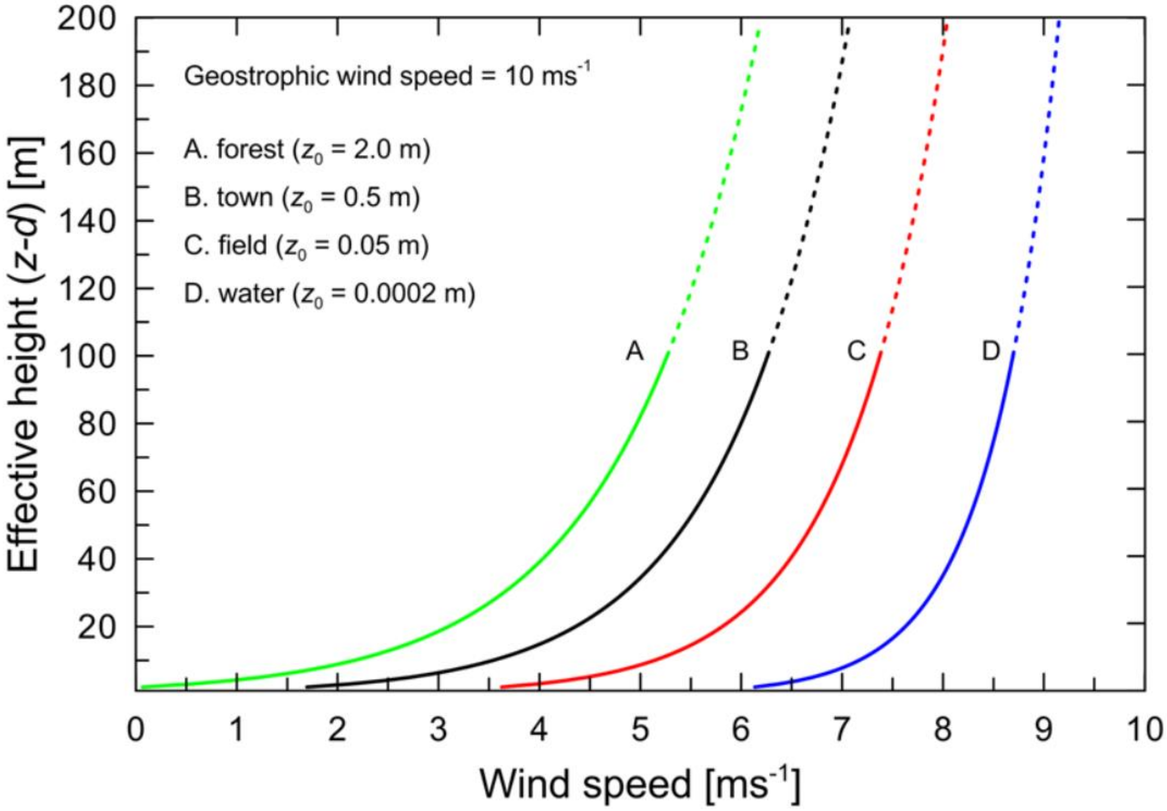


Figure 2-16 shows how the wind profile gets affected by the roughness[57].

3 Method For Modelling

This chapter will describe how the wind turbines and wind farms were models. It will also include how information about wind conditions were gathered and used in the wind farm models. To make the models as accessible, user friendly and PyWake compatible as possible, the Hornsrev1 offshore wind farm model was used as a basis. This model is used in the PyWake QuickStart tutorial, and the models made in this thesis will therefore be compatible with PyWake and all the functions used in the tutorial.

3.1 Wind Turbine Positions

In PyWake the turbine positions or the layout of a wind farm is described with the x and y coordinates of each turbine. To model new wind farms a screenshot of the specific wind farm was taken from the 4COffshore global offshore map[29]. Then the scale indicator on the map was used to scale the length of one pixel. When the length of one pixel was calculated a reference point on the wind farm was decided on, this was used as the (0, 0) point of coordinates for the wind farm. Then the distance from the reference point to each wind turbine were measured, and the pixel x and y coordinates were found in relation to the reference point. These coordinates were then scaled into coordinates in meters using the ratio of meters per pixel found from the scale indicator. Figure 3-1 shows the screenshot from 4COffshore used find coordinates for the Borkum Riffgrund II wind farm with each turbine numbered. When modelling the Borkum Riffgrund II wind farm the turbine numbered 2 in Figure 3-1 was chosen as the reference point with coordinates (0, 0). The scale indicator used is shown in the bottom right of Figure 3-1. These x and y coordinates were then used in the same way as in the Hornsrev1 reference model to describe the layout of the wind farm. The tool used to find coordinates was the online pixel ruler from RapidTables[58].

Figure 3-1 shows screenshot of Borkum Riffgrund II from 4COffshore with each turbine numbered.

3.2 Wind Turbines

The wind turbines are described in the same way as in the reference Hornsrev1 reference wind farm model file. The power curves of the wind turbines are specified, for the models modelled in this thesis the power curves shown in chapter 2.1 are used. The hub height, diameter and thrust coefficient for the wind turbines at each wind farm is also specified here. As the thrust coefficient of the turbines modelled in this thesis is not publicly available, reference thrust coefficients curves was used. PyWake also allows for input such as gear losses, generator losses and conversion losses of the turbine, as this information wasn't available the default values used in PyWake have been used.

This information is combined and named after the wind turbine it is based on to form the wind turbine models. The wind turbine models can also be imported to other wind farms.

3.3 Wind Conditions

To model the wind conditions at each wind farm, the coordinates for the wind farm was found. These coordinates were then chosen in the Global Wind Atlas map. The data from the wind frequency rose was collected and used to define the wind direction probability for the different sites. Weibull scale and shape parameters were found when downloading the Generalized Wind Climate file for the location, parameters from the closest height to the hub height of the wind turbines at the wind farm was used. The parameters where chosen for references roughness of 0.00 meters as that was the closest to the roughness over water (0.0002 m) shown in Figure 2-16.

3.4 Estimating Annual Energy Production

To estimate the annual energy production for each of the wind farms modelled, the simulationResult() object and the NOJ wake deficit model was used as shown in the PyWake Quickstart tutorial[59]. Figure 3-2 shows an example of how to estimate the AEP of the Borssele III and IV wind farm model by following the PyWake tutorial.

```
from py_wake.examples.data.Borssele3and4 import Borssele3and4, wt_x, wt_y, V164
from py_wake import NOJ

#Importing the site, windturbine type and the wake deficit model to be used.
site = Borssele3and4()
windTurbines = V164()
noj = NOJ(site,windTurbines)

simulationResult = noj(wt_x,wt_y)
simulationResult.aep
print ("Total AEP: %f GWh"%simulationResult.aep().sum())
```

Figure 3-2 shows how to estimate annual energy production on the Borssele III and IV wind farm by following the PyWake tutorial.

The files made for each wind farm modelled in this thesis contains the necessary commands to estimate the AEP and wake losses by running the file.

4 Results

In this chapter the different wind farm models will be presented, and what parameters and data is used in the models will be shown. The estimated AEP for each wind farm will be presented and compared with data from the actual wind farm where that is possible. It will also be shown that the wind farm models are compatible with different PyWake functions.

The file can be found on GitHub in the repository called Offshore wind farms for PyWake or by the following link: https://github.com/StianJS/Offshore-wind-farms-for-PyWake

4.1 Borssele III & IV

Here the model based on the Borssele III and IV wind farm will be presented, what parameters that were used and the estimated energy production will be compared to the electricity production claimed by Blauwind. The code for the Borssele III & IV wind farm is shown in Appendix A and the file is uploaded to GitHub.

4.1.1 Layout and wind conditions

The coordinates for the 77 turbines of the wind farm were mapped out as described in 3.1 and the x and y coordinates entered in the Python file. Figure 4-1 shows a screenshot of the wind farm from the 4COffshore global offshore map where each turbine has been numbered, and a map plotted of the wind farm model in python. The numbers of each turbine in the python plot is one number higher compared to the turbine on the map from 4COffshore as Python starts counting from 0. The wind turbine numbered 21 on the 4COffshore map and 20 in the map plotted on python was chosen as the (0,0) point for the coordinates.

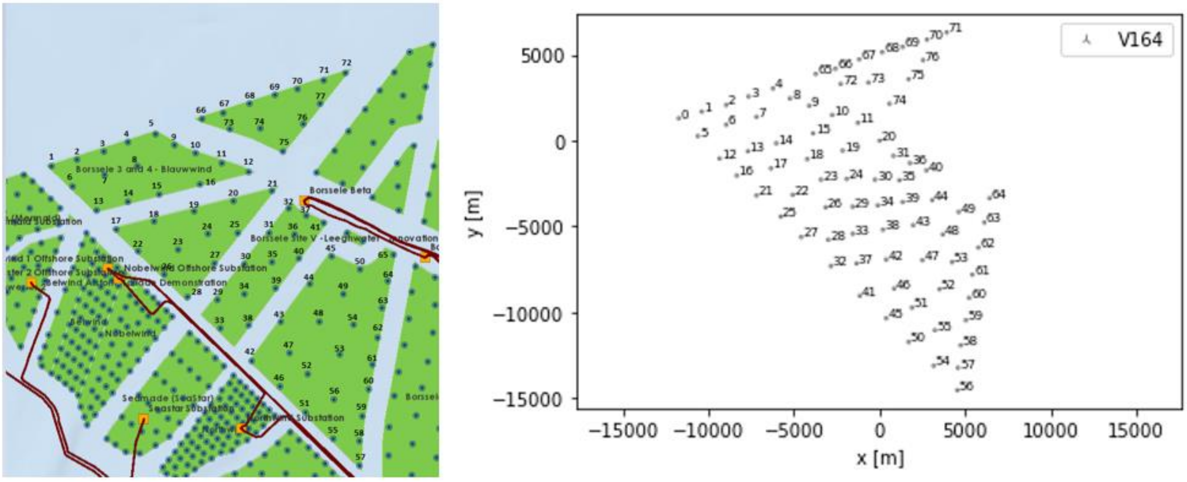


Figure 4-1 show a map of the Borssele II and IV windfarm from the 4COffshore global offshore map, and the turbine layout of the model plotted in Python.

The data from GWA at the coordinates of the wind farm, N 51.438601, E 3.026184, was used. Data from the height of 100 meters were used as that was the closest to the hub height of 109 meters of the wind turbine. Figure 4-2 show the wind frequency rose at the site made from data from GWA and plotted in Python. Table 4-1 show the wind data and parameters used to model the wind conditions at the wind farm.

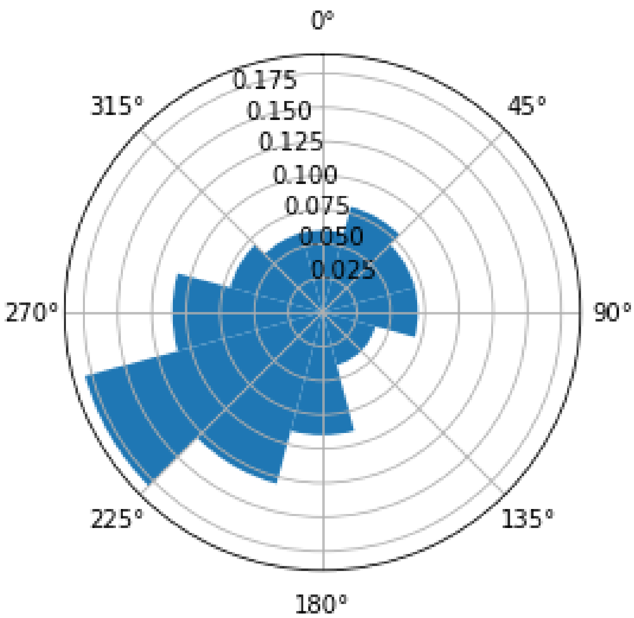


Figure 4-2 shows the wind frequency rose made from data at the coordinates of the wind farm at a height of 100 meters.

Table 4-1 shows the wind data used for the Borssele III & IV windfarm model.

Sector	1	2	3	4	5	6	7	8	9	10	11	12
Wind Frequency [%]	6	8	7	7	4	4	9	13	18	11	7	6
Weibull A	8,60	8,89	9,52	9,81	9,35	8,90	12,26	12,08	12,57	9,96	8,95	9,05
Weibull K	2,213	2,400	2,732	2,639	3,014	2,311	2,592	2,736	2,482	2,068	1,889	1,979

4.1.2 V164-9.5 MW Wind Turbine

The power curve shown in 2.1.2 was implemented along with a height of 109 meters. The diameter of the wind turbine was set to 164 meters. As the thrust coefficient of the V164-9.5 MW turbine wasn't available, the thrust coefficient curve of the reference DTU10MW turbine was used[13]. Figure 4-3 shows the power and thrust coefficient curve of the V164 turbine in Python.

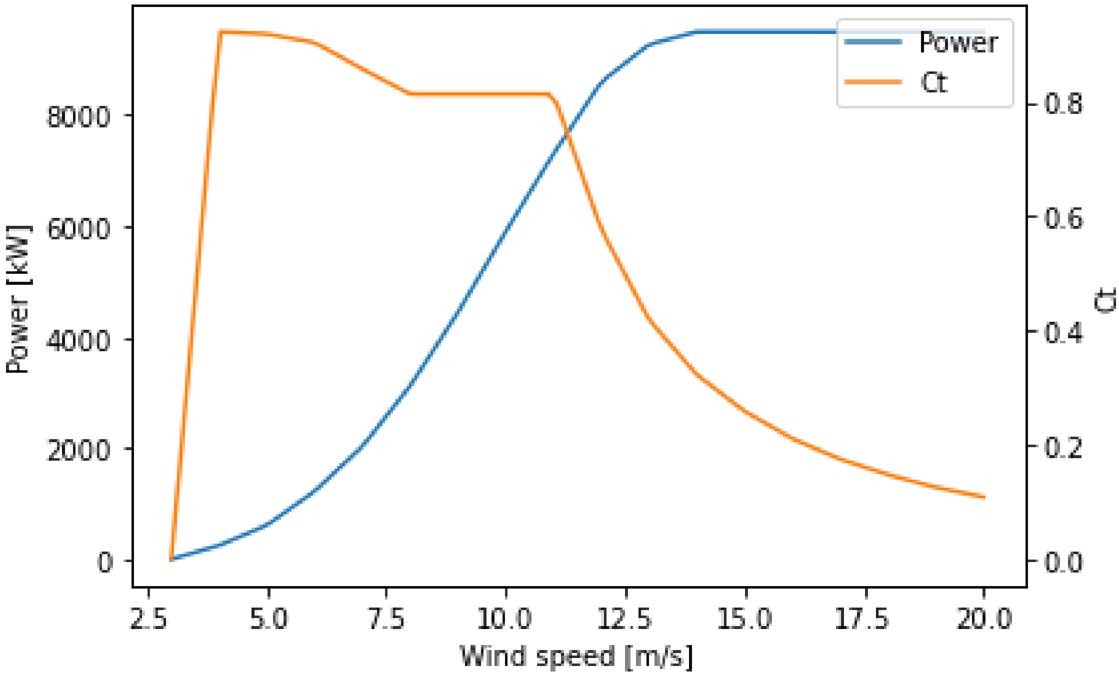


Figure 4-3 shows the plotted power and thrust coefficient curve of the V164-9.5 MW turbine modelled.

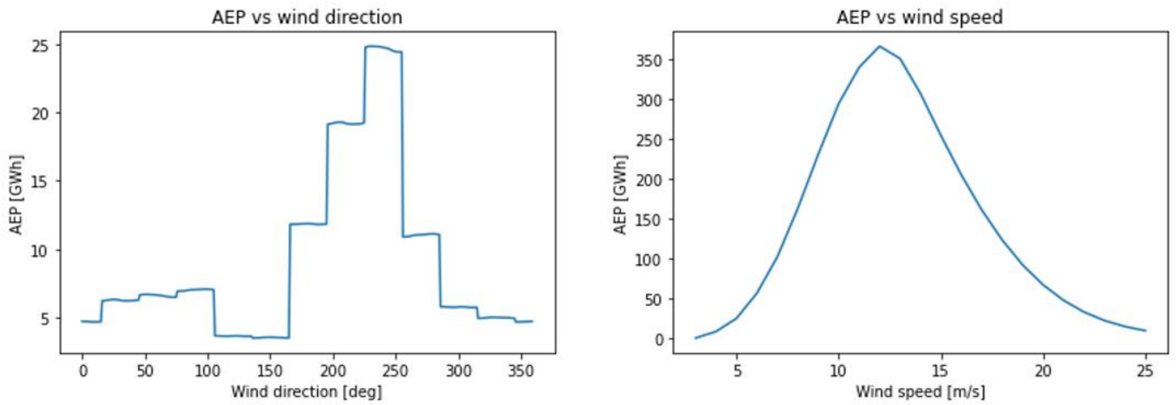
4.1.3 Estimated AEP and wake losses

When using the NOJ wake deficit model on the model of the wind farm with the V164-9.5 MW wind turbine the total AEP is shown in Table 4-2 and was estimated to be 3 103.29 GWh. The wake losses was estimated to be 92.71 GWh in a year.

Table 4-2 shows the AEP according to Blauwind and the estimated AEP.

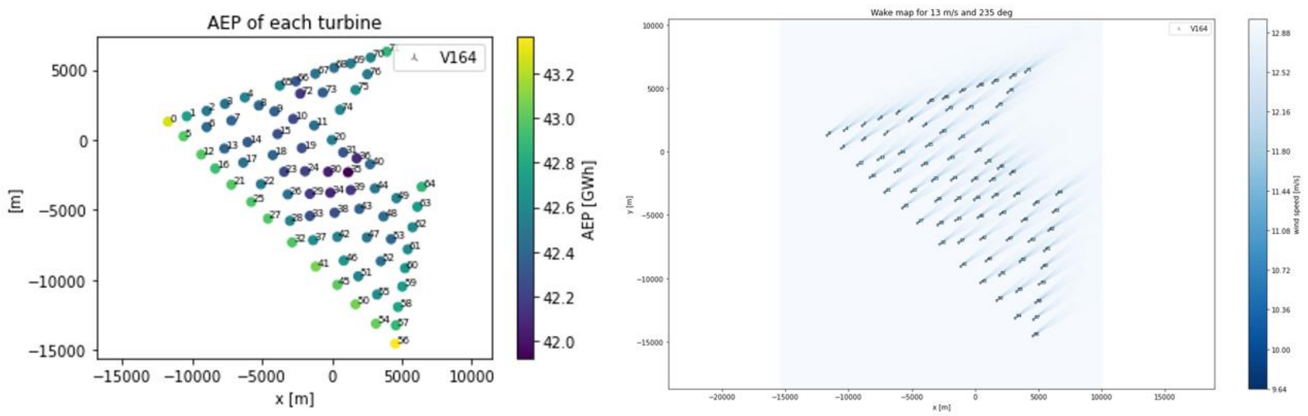
AEP according to Blauwind	Estimated AEP	Difference
3 000 GWh	3 103.29 GWh	+3.44%

Figure 4-4 (a) shows the estimated AEP plotted against the wind direction, while (b) shows the estimated AEP plotted against the wind speed. Figure 4-5 (a) shows the estimated AEP of each wind turbine while Figure 4-5 (b) shows the wake flow map simulated for a wind speed of 13 m/s and wind direction 235 degrees using the NOJ wake deficit model.



(a) (b)

Figure 4-4, (a) shows the AEP plotted against the wind direction. (b) shows the AEP plotted against the wind speed.



(a) (b)

Figure 4-5 (a) show the estimated AEP of each wind turbine, (b) show the wake flow map for wind conditions: wind speed = 13 m/s and wind direction from 235 degrees.

4.2 Borssele I and II

Here the model of the Borssele I and II offshore wind farm will be presented. What parameters used will be shown and the estimated AEP will be compared to the claimed AEP. The code for the Borssele I & II wind farm is shown in Appendix B and the file is uploaded to GitHub.

4.2.1 Layout and wind conditions

The coordinates of the 94 wind turbines were mapped as described in 3.1 and the results are plotted and shown in Figure 4-6. As the Borssele I and II wind farm is located with the Borssele III and IV wind farm it was chosen to use the same point as the (0, 0) reference point, this makes it possible to combine the two wind farms for simulations of the entire Borssele wind farm zone.

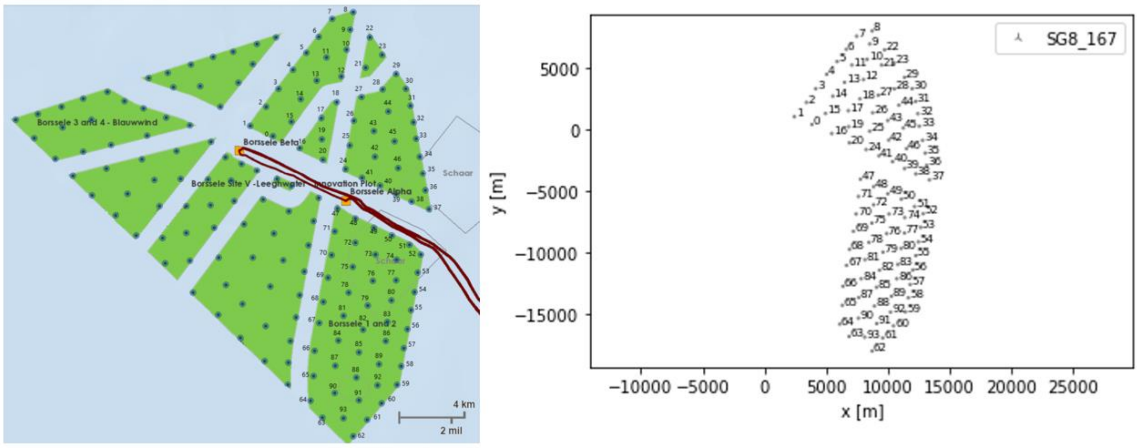


Figure 4-6 show map of Borssele I and II from 4COffshore and the model plotted in python.

The data from GWA at the coordinates of the wind farm, N 51.438601, E 3.026184, the same as for the Borssele III and IV model. The parameters of height of 100 meters was used as that is the closest to the hub height of 116 meters. The wind frequency rose was the same and shown in Figure 4-2 but A and K parameters differed. The parameters used are shown in Table 4-3.

Table 4-3 parameters for modeling wind at Borssele I and II at 100 meters height.

Sector	1	2	3	4	5	6	7	8	9	10	11	12
Wind Frequency [%]	6	8	7	7	4	4	9	13	18	11	7	6
Weibull A	8,60	8,89	9,52	9,81	9,35	8,90	12,26	12,08	12,57	9,96	8,95	9,05
Weibull K	2,213	2,400	2,732	2,639	3,014	2,311	2,592	2,736	2,482	2,068	1,889	1,979

4.2.2 SG 8.0-167 DD wind turbine

Information from 2.1.3 was implemented. Hub height was set to 116 meter and diameter to 167 meters. Figure 4-7 show the power and Ct curve used. As the Ct curve of the turbine wasn't available the Ct curve from reference LW 8 MW turbine was used.

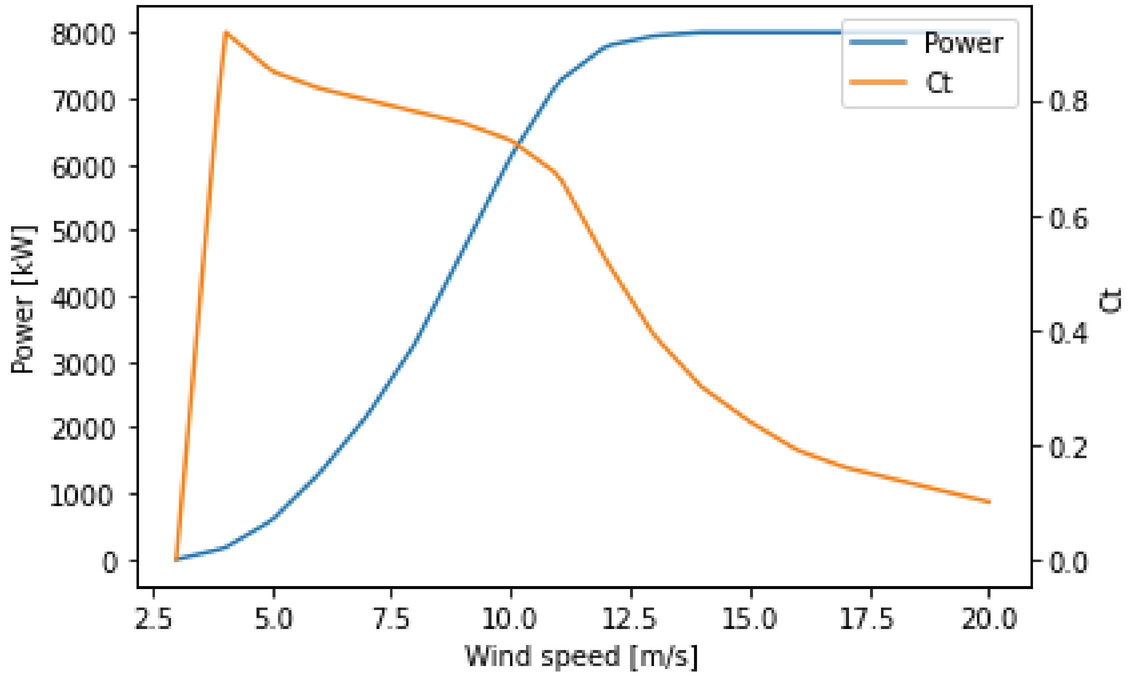


Figure 4-7 show the power and Ct curve of the SG 8.0-167 DD wind turbine model.

4.2.3 Estimated AEP and wake losses

When using the NOJ wake deficit model on the model of the wind farm with the SG 8.0-167 DD wind turbine the total AEP is shown in Table 4-4 and was estimated to be 3 522.41 GWh. The losses due to wake was estimated to be 118.99 GWh in a year.

Table 4-4 show the stated AEP and the estimated AEP of the Borssele I and II wind farm.

AEP according to Ørsted	Estimated AEP	Difference
Around 3 000 GWh	3 522.41 GWh	+17.41%

The figures Figure 4-8 and Figure 4-9 are plotted like Figure 4-4 and Figure 4-5 to show that the wind farm model works with different PyWake functions. Figure 4-8 (a) shows the estimated AEP plotted against the wind direction, while (b) shows the estimated AEP plotted against the wind speed. Figure 4-9 (a) shows the estimated AEP of each wind turbine while

Figure 4-9 (b) shows the wake flow map simulated for a wind speed of 13 m/s and wind direction 235 degrees using the NOJ wake deficit model.

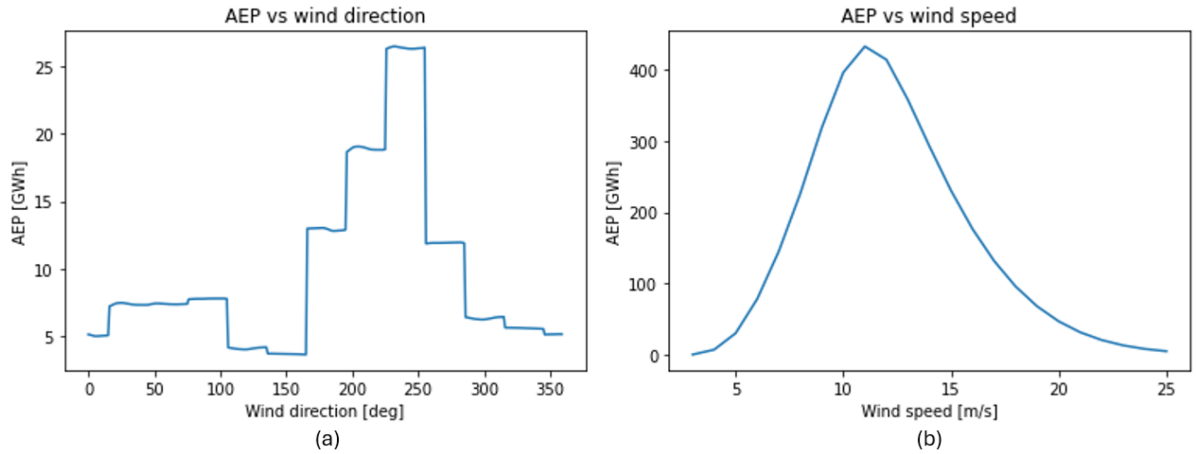


Figure 4-8 (a) shows the AEP plotted against the wind direction. (b) shows the AEP plotted against the wind speed.

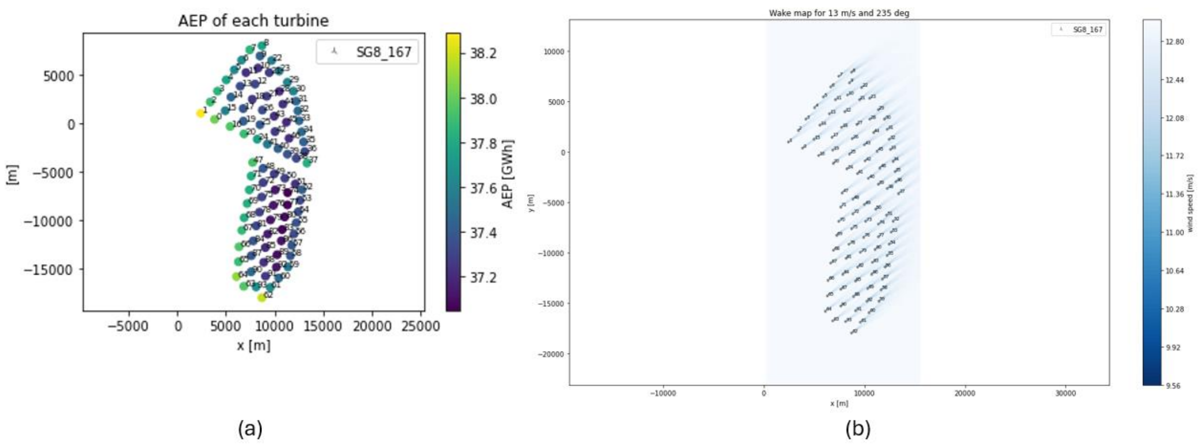


Figure 4-9 show the estimated AEP of each wind turbine, (b) show the wake flow map for wind conditions: wind speed = 13 m/s and wind direction from 235 degrees.

4.3 Borssele wind farm zone

As the models for Borssele I and II and Borssele III and IV have been made with the same reference point it is possible to combine the layouts and make a model for the entire Borssele wind farm zone. Figure 4-10 shows the layout when the two models have been combined, with the V164 wind turbine and the SG8-167 wind turbine. The site uses parameters at 100 meters shown in Table 4-3 for the SG8.0-167 DD turbines and the V164 turbines. When estimating the AEP of the Borssele wind farm zone using the NOJ wake deficit model, the estimated AEP shown in Table 4-5 is 6 409.13 GWh. The estimated wake loss is 215.11 GWh in a year. In Figure 4-11 the wake flow map simulated for the entire Borssele wind farm zone for a case with wind speeds of 13 m/s and wind direction 235 degrees. The code for the Borssele wind farm zone is shown in Appendix C and the file is uploaded to GitHub.

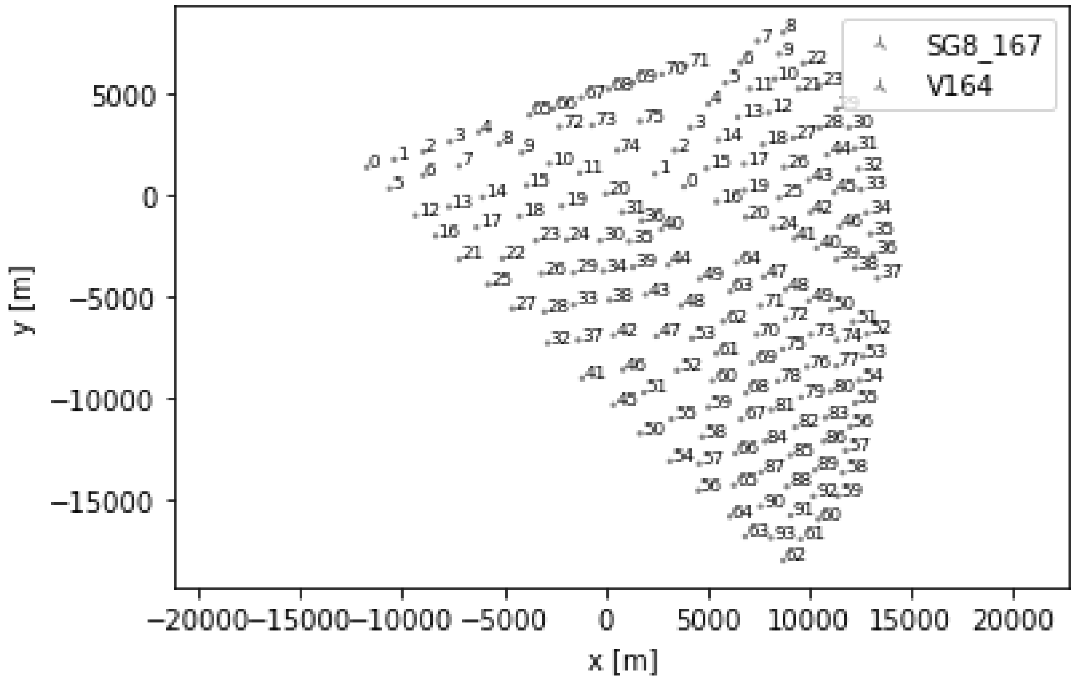


Figure 4-10 shows the layout of the Borssele wind farm zone.

Table 4-5 show the stated AEP of the Borssele wind farm zone and the estimated AEP.

AEP of Borssele I and II and Borssele III and IV combined	Estimated AEP	Difference
Around 6 000 GWh	6 409.13 GWh	+6.82%

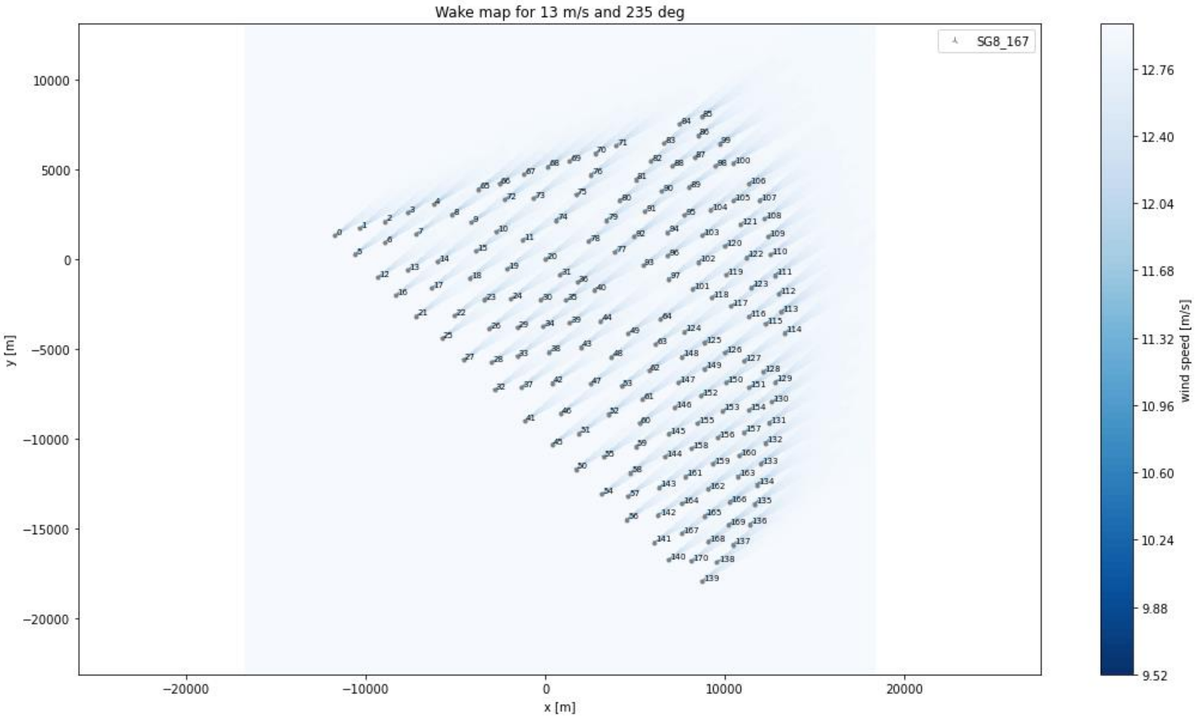


Figure 4-11 show the wake flow of a case of the Borssele wind farm zone with wind speeds of 13 m/s and wind direction of 235 degrees.

4.4 Borkum Riffgrund II

In this chapter the model based on the Borkum Riffgrund II wind farm will be presented, what parameters that were used and the estimated energy production will be compared to the actual production. The code for the Borkum Riffgrund II wind farm is shown in Appendix D and the file is uploaded to GitHub.

4.4.1 Layout and wind conditions

Coordinates of the 56 wind turbines was found as described in 3.1, Figure 4-12 show the wind turbine layout from the 4COffshore map and the position of the turbines plotted in python.

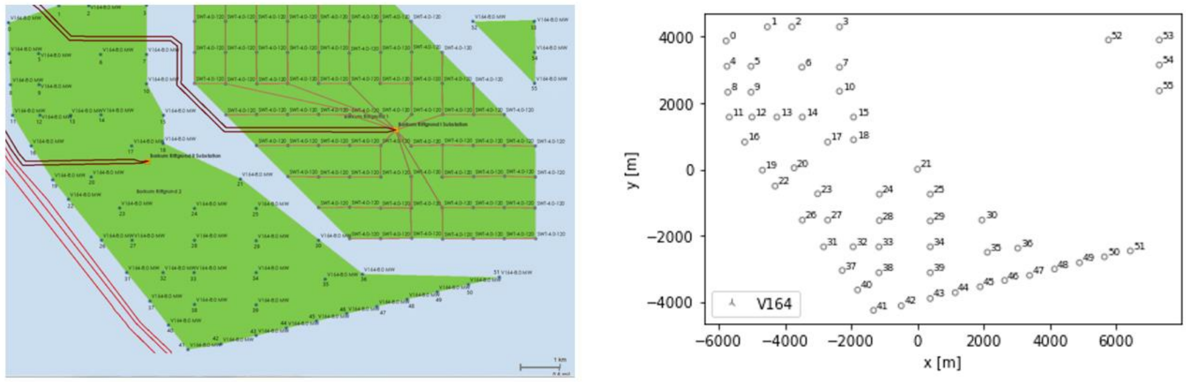


Figure 4-12 show overview of the wind farm in 4COffshore and the positions of turbines plotted in Python.

The data from GWA at the coordinates of the wind farm, N 54,580061, E 6,294479, was used. Data from the height of 100 meters were used as that was the closest to the hub height of 105 meters of the wind turbine. Table 4-6 shows the wind data and Weibull parameters used for the model.

Table 4-6 shows the wind data and Weibull parameters used for the model.

Sector	1	2	3	4	5	6	7	8	9	10	11	12
Wind Frequency[%]	5	4	6	8	6	6	8	12	14	12	10	9
Weibull A	8,48	7,56	9,29	11,1	11,54	10,99	11,28	12,67	12,67	11,96	11,54	10,26
Weibull K	2,166	2,162	2,178	2,299	2,604	2,607	2,002	2,467	2,396	2,342	2,568	2,186

4.4.2 V164-8 MW Wind Turbine

The information and data from 2.1.1 was used to make the model. Hub height was set to be 105 meters, diameter to 164 meters and the power and thrust coefficient used is shown in Figure 4-13. As the thrust coefficient curve of the V164-8 MW turbine wasn’t available the curve of the LW 8 MW reference turbine based on the V164-8 MW turbine was used.

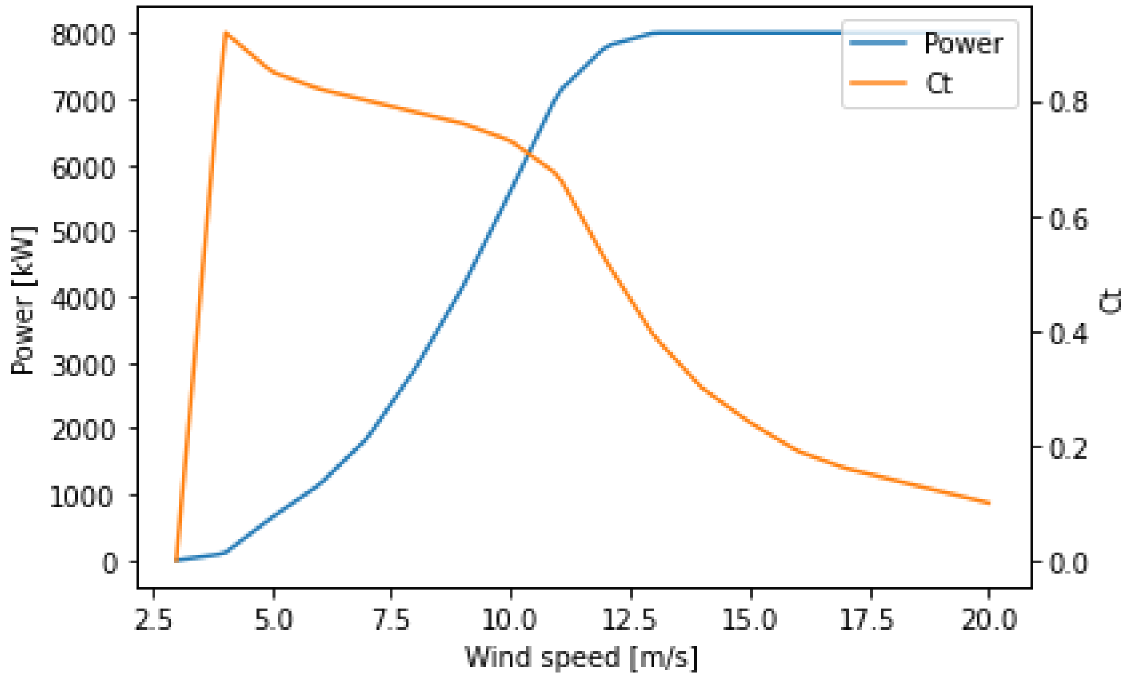


Figure 4-13 shows the power and thrust coefficient curve of the V164-8 MW wind turbine model plotted in Python.

4.4.3 Estimated AEP and wake losses

When using the NOJ wake deficit model on the model of the wind farm with the V164-8 MW wind turbine the total AEP is shown in Table 4-2 and was estimated to be 2 154.80 GWh. The annual electricity production at Borkum Riffgrund II between 2019 and 2023 was shown in Table 2-2 and ranged between 1 165 to 1 450 GWh. Annual wake losses was estimated to be 97.3 GWh.

Table 4-7 shows the AEP according to Blauwind and the estimated AEP.

AEP between 2019 and 2023	Estimated AEP	Difference
1 165 – 1 450 GWh	2 154.80 GWh	+48.60-84.96%

Figure 4-14 (a) show the estimated AEP plotted against the wind direction while Figure 4-14 (b) show the estimated AEP plotted against the wind speed. Figure 4-15 (a) show the estimated AEP of each of the turbines while Figure 4-15 (b) show the estimated wake flow map of a case with wind speeds of 13 m/s and a wind direction of 235 degrees.

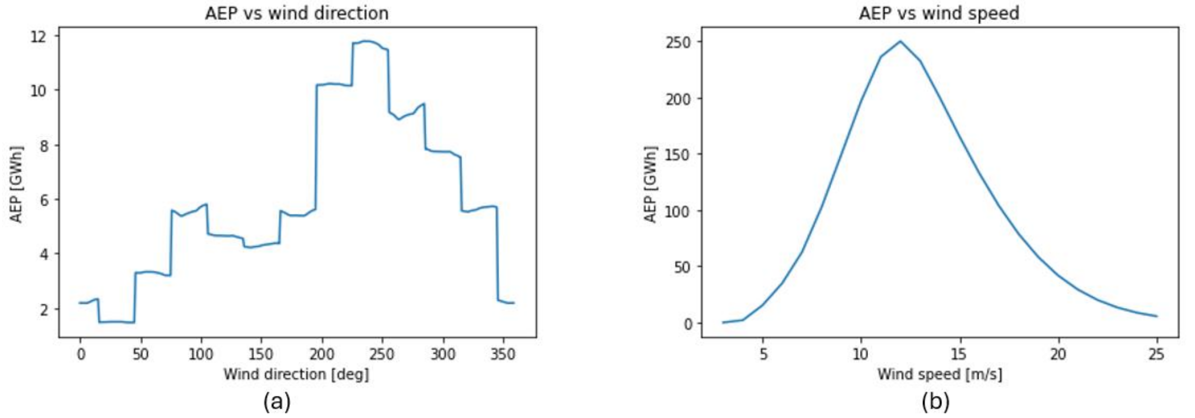


Figure 4-14 (a) shows the AEP plotted against the wind direction. (b) shows the AEP plotted against the wind speed.

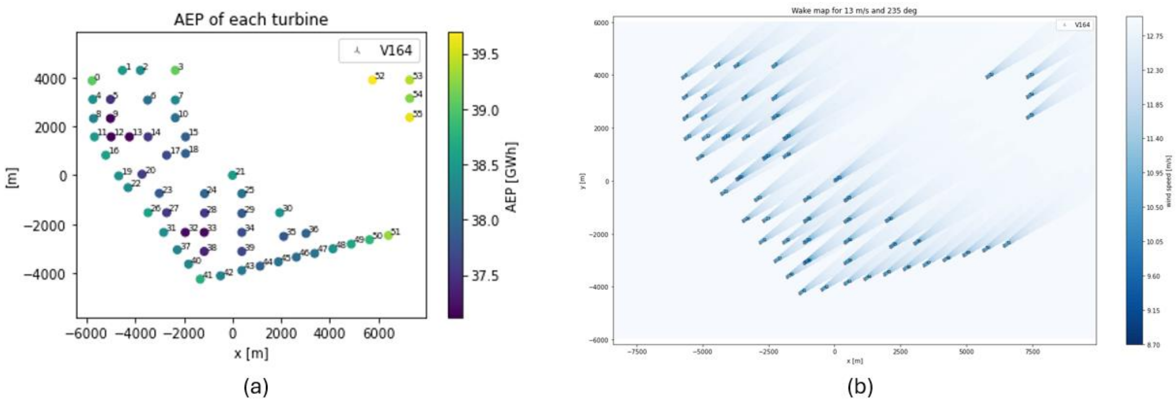


Figure 4-15 (a) show the estimated AEP of each turbine, (b) show the estimated wake flow map for wind conditions wind speed =13 m/s from wind direction 235 degrees.

4.5 Hornsea 2

Here the model based on the Hornsea 2 wind farm will be presented, what parameters that were used and the estimated energy production will be compared to the actual production. The code for the Hornsea 2 wind farm is shown in Appendix E and the file is uploaded to GitHub.

4.5.1 Layout and wind conditions

Coordinates for each of the 165 turbines were found and the results are shown in Figure 4-16 with the wake flow map plotted for a case with wind speed of 13 m/s and wind direction 285 degrees to show the model is compatible with running PyWake simulations.

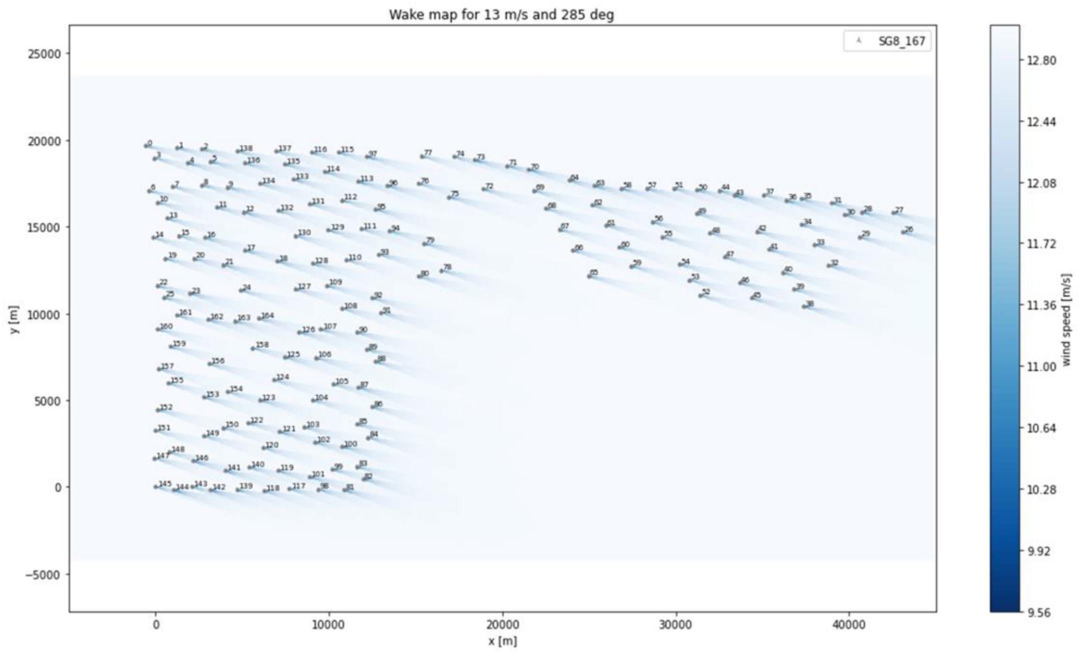


Figure 4-16 show the layout of the model of Hornsea 2 with the wake flow map plotted for a case wind wind speed of 13 m/s and wind direction 285 degrees.

Data and parameters used were from coordinates N 53.530064 and E 1.472992 at the GWA. The parameters for height 100 meters were used as that was closest to the hub height of 116 meters. Table 4-8 show the parameters used to model the wind conditions at the Hornsea 2 site at 100 meters.

Table 4-8 show parameters used to model wind conditions at Hornsea 2 site at 100 meters.

Sector	1	2	3	4	5	6	7	8	9	10	11	12
Wind Frequency[%]	6	4	5	7	5	8	10	13	16	11	8	7
Weibull A	9.74	8.58	9.03	10.06	9.08	10.50	11.65	13.18	12.08	11.95	10.04	10.25
Weibull K	2.557	2.279	2.607	2.232	2.037	2.506	2068	2.428	2.760	2.256	2.471	2.182

4.5.2 Wind turbine

The wind turbine model used for the Hornsea 2 site is the same as described and shown in 4.2.2. Based on the SG 8.0-167 DD wind turbine with a height of 116 meters and diameter of 167 meters.

4.5.3 Estimated AEP and wake losses

When using the NOJ wake deficit model on the model of the wind farm with the SG8.0-167 DD wind turbine the total AEP is shown in Table 4-9 and was estimated to be 6 670.45 GWh compared to the 2023 electricity production of 5 978 GWh. The annual wake loss was estimated to be 161 GWh.

As with the previous models the Hornsea 2 model have been tested with different PyWake commands and functions to make sure the model is compatible and can run PyWake simulations in an easy way.

Table 4-9 show electricity produced at Hornsea 2 in 2023 and the estimated AEP.

Electricity produced in 2023	Estimated AEP	Difference
5 978 GWh	6 670.45 GWh	+11.58%

5 Discussion And Uncertainties

When making the models shown in chapter 4, information and data from different publicly available sources was used. And where information wasn't available such as with thrust coefficient curves, reference data was found and used. In the case of using the reference LW 8 MW Ct curve for the Vestas V164-8 MW, the uncertainty is smaller as the reference LW 8 MW wind turbine was based on data from the V164-8 MW turbine and independently validated. While the SG 8.0-167 DD and V164-9.5 MW turbine models used Ct curves from reference turbines at similar size but not based on them. The models also used the default parameter defined in PyWake such as generator loss, generator loss and converter loss as this information wasn't available for the turbines.

The parameters used to describe the wind conditions at each site provided from Global Wind Atlas are models and calculated values based on actual measurements. And according to Global Wind Atlas itself the Globa Wind Atlas should be used to perform preliminary calculations and as a starting point before commissioning further studies including wind measurements for specific sites[53, 55]. This means there are uncertainty in the accuracy for the parameters describing the wind at the sites.

When making the layout of the wind farms, there will be some smaller deviations in the turbines positioning in the model compared to the actual positions. This is due to scaling up the pixel coordinates, the length of one pixel in the maps used to model varied from 13.7 to 40 meters at the different farms. The models made in this thesis does not factor in neighboring wind farms and the wake effects from them. From 2.3 we know wake from wind farms often reaches 15 km downwind but have even been observed up to 55 km downwind. As seen in Figure 4-1 there are other wind farms located close to the Borssele wind farm zone, this is true also for Borkum Riffgrund II and Hornsea 2. Wake effects from the farms have not been considered when estimating the AEP of wind farms modeled in this thesis.

As described in chapter 2.4.2 the NOJ wake deficit model makes simplifications and ignore near wake in its calculations. This contributes to uncertainty in the estimated wake effects, and when the estimated AEP is being compared to the actual AEP. It was also found in "Comparing wind farm production data to engineering wake model simulations" by M. al Halabi that the NOJ wake deficit model estimated higher AEP than other wake deficit models in PyWake when using the same parameters. The NOJ model estimated a 23.89% higher AEP than the actual production at Horns Rev 1 in the study. For all the models in this thesis the estimated AEP with the NOJ wake deficit model was higher than stated or actual AEP, other wake deficit models would therefor most likely produce lower estimated AEP closer to the stated or meassured AEP.

The Borssele I & II and Borssele III & IV wind farms were made with the same (0,0) coordinates and wind conditions, this made it possible to combine the two different wind farms to model the entire Borssele wind farm zone. The results from this were shown in chapter 4.3. Here we can see that the calculated AEP for the Borssele wind farm Zone is lower than the each of the calculated AEP from the two wind farms combined when simulated on their own. How combining the two wind farms effect the estimated production from each turbine is some that can be investigated in the future.

From the electricity production data from Borkum Riffgrund II we can see that the annual electricity production of a wind farm varies from year to year, which show that production data from only one year can be misleading. This makes comparing the estimated AEP with the measured AEP of newer wind farms, such as Hornsea 2 which opened in 2022 and only have one full year of electricity production to use for comparison uncertain. As production this year could be an outlier compared to the average annual electricity production over a longer period.

The difference in estimated AEP and actual or stated AEP ranged from 3.44 to 17.41% higher for the models except the Borkum Riffgrund II model. For the Borkum Riffgrund model the difference was between 48.60 and 84.96% depending on which year of production it was compared with. As the turbines at the Borkum Riffgrund II wind farm is closer than the turbines at the other wind farms modelled, the limitations of the NOJ wake deficit model might play a role. As described in 2.4.2, the NOJ wake deficit model makes simplifications for the near wake. These simplifications can be causing the wake in the simulations to develop faster than the actual wake. While in the other wind farms where the turbines are further apart the difference in wake development might be smaller. Borkum Riffgrund II is also located close to other wind farms, the wake from them are not taken into considerations for the simulations. And this can be a cause contributing to the higher simulated AEP than the actual AEP. How big of a role these factors play can be investigated in future studies.

In some of the wind farms modeled, Borssele 1 & 2 and Borssele 3 & 4 only information about the stated AEP from the owners of the wind farms was possible to find. Not the actual production numbers. This contributes to uncertainty in the comparisons as the exact numbers aren’t known.

6 Conclusion

The goal of this thesis was to gather enough information and data from public sources to get good descriptions of different offshore wind farms. Then use this information to make models of theses wind farms where the data is accessible and can easily be used for running PyWake simulations. The final part of the task for this thesis was if time allowed it to run simulations of the wind farm models and compare the simulated production with the actual production. This has been done for all wind farms modelled in this thesis.

Data and information about offshore wind farms was collected from different public sources to get an as good as possible description of offshore wind farms. This included information regarding the different turbines and the properties of the turbines at the wind farms. Where information such as Ct curves wasn't publicly accessible, data from reference turbines of similar size was used. In the case of the V164-8 MW wind turbine, the reference wind turbine used to find Ct curve was based on the V164-8 MW and the reference wind turbine had been independently validated. Information regarding losses in mechanical components such as generator and gears wasn't publicly available so the default values in PyWake have been used. The wind turbines modelled in this thesis are the Vestas V164-8 MW wind turbine, Vestas V164-9.5 MW wind turbine and the Siemens Gamesa SG 8.0-167 DD wind turbine.

Layout of the offshore wind farms have been modelled from the 4COffshore global offshore map to create the coordinates for the turbines used in the models for each wind farm. Every wind farm was modelled as a uniform Weibull site, which is a site with uniform winds where the probability of each wind direction is specified. With Weibull distributed wind speeds. The data for the wind conditions have been gathered from Global Wind Atlas at the location of each wind farm. As stated by Global Wind Atlas this data is estimations and should be used for initial or preliminary calculations, while actual measurements at the locations are recommended for more accurate data. The wind farms modelled in this thesis are Borssele I & II, Borssele III & IV, the Borssele wind farm zone, Borkum Riffgrund II and Hornsea project 2.

The NOJ wake deficit model was used in this thesis to estimate the AEP of each wind farm model. As in previous work using the NOJ model, the estimated AEP when using the NOJ model was higher than the actual AEP for the wind farms. When compared the estimated AEP was 3.44% higher at Borssele III & IV, 17.41% higher at Borssele I & II, 6.82% higher for the Borssele wind farm zone, 11.68% higher for the Hornsea project 2 wind farm. For the simulations of Borkum Riffgrund 2 the simulated AEP was between 48.60-84.96% higher depending on which year of production it was compared with.

The different models made in this thesis are compatible and capable of running PyWake simulations as shown in chapter 4, the models are made in such a way that they can be used by following the PyWake QuickStart tutorial. In the case of the Borssele wind farm zone it was also shown that two models with different turbines could be combined.

References

[1] IEA, "World Energy Outlook 2019," 2019. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2019

[2] WFO, "Global Offshore Wind Report," 2024. [Online]. Available: https://wfo-global.org/wp-content/uploads/2024/04/WFO-Report-2024Q1.pdf

[3] ERM. "Report finds most countries expected to miss net zero offshore wind targets." ERM. https://www.erm.com/news/report-finds-most-countries-expected-to-miss-net-zero-offshore-wind-targets/ (accessed 2024).

[4] L. Fernández, "Share of wind electricity generation worldwide 2010-2022," 2023. [Online]. Available: https://www.statista.com/statistics/1302053/global-wind-energy-share-electricity-mix/

[5] Ember. "Share of electricity generated by wind power." https://ourworldindata.org/grapher/share-electricity-wind (accessed 2024).

[6] KEBA. "Pitch control and yawing: systems for optional wind turbine design." https://www.keba.com/en/news/industrial-automation/pitch-and-yaw-systems-wind-turbine (accessed 2024).

[7] The_Wind_Power. "V164/8000." https://www.thewindpower.net/turbine_en_867_vestas_v164-8000.php (accessed Febuary, 2024).

[8] wind-turbine-models. "MHI Vestas Offshore V164-8.0MW." https://en.wind-turbine-models.com/turbines/1419-mhi-vestas-offshore-v164-8.0-mw (accessed March, 2024).

[9] Vestas. "Our Global Track Record." https://us.vestas.com/en-us/products/track-record#accordion-562c3c810a-item-91991e0e8c (accessed April, 2024).

[10] J. M. Cian Desmond, Lindert Blonk, Wouter Haans, "Description of an 8 MW reference wind turbine," in *Conference on Science of Making Torque from Wind (TORQUE)*, Munich, Germany, 2016, vol. Journal of Physics: Conference Series Volume 753, no. 9: IOP Publishing Ltd, doi: 10.1088/1742-6596/753/9/092013. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/753/9/092013#references

[11] Vestas. "V164-9.5 MW." https://us.vestas.com/en-us/products/offshore/V164-9-5-MW (accessed Febuary, 2024).

[12] The_Wind_Power. "V164/9500." https://www.thewindpower.net/turbine_en_1476_vestas_v164-9500.php (accessed Febuary, 2024).

[13] F. Z. Christian Bak, Robert Bitsche, Taeseong Kim, Anders Yde, Lars Christian Henriksen, Morten Hartvig Hansen, José Pedro Albergaria Amaral Blasques, Mac Gaunaa, Anand Natarajan, "The DTU 10-MW Reference Wind Turbine," presented at the Danish Wind Power Research 2013, Trinity, Fredericia, Denmark, 2013, online. [Online]. Available: https://orbit.dtu.dk/en/publications/the-dtu-10-mw-reference-wind-turbine.

[14] The_Wind_Power. "SG 8.9-167 DD." https://www.thewindpower.net/turbine_en_1558_siemens-gamesa_sg-8.0-167-dd.php (accessed March, 2024).

[15] *Providing more predictable power output*, 2012. [Online]. Available: https://pdf.archiexpo.com/pdf/siemens-gamesa/high-wind-ride-through/88089-134501.html. Accessed: 2024.

[16] S. Gamesa. "SG 8.0-167 DD Offshore wind turbine." https://www.siemensgamesa.com/products-and-services/offshore/wind-turbine-sg-8-0-167-dd (accessed March, 2024).

[17] Power_Technology. "Borssele III and IV Offshore Wind Farm." 2024. https://www.power-technology.com/projects/borssele-iii-iv-offshore-wind-farm/ (accessed.

[18] Netherlands_Enterprice_Agency, "wind & water works," 2022. [Online]. Available: https://english.rvo.nl/sites/default/files/2021/10/Dutch%20Offshore%20Wind%20Guide%202022.pdf

[19] Blauwind. "Delivering 731.5 MW of renewable energy." https://www.blauwwind.nl/en/ (accessed 2024).

[20] Blauwind. "About the project." https://www.blauwwind.nl/en/about/about-blauwwind (accessed 2024).

[21] Dutch_Government. "Kavelbesluit IV windenergiegebied Borssele." https://wetten.overheid.nl/BWBR0037800/2016-09-03 (accessed 2024).

[22] W. maps, "Wind farm layout," ed.

[23] Ørsted. "Ørsted's first Dutch offshore wind farm fully commissioned." https://orsted.com/en/media/news/2020/11/448070886682487 (accessed 2024).

[24] J. Lindner. "Statistics About The Average Energy Consumption Per Household." https://gitnux.org/average-energy-consumption-per-household/ (accessed 2024).

[25] "Borssele I and II Offshore Wind Power Project." NS Energy. https://www.nsenergybusiness.com/projects/borssele-i-and-ii-offshore-wind-power-project/ (accessed 2024).

[26] N. E. Agencu. "Borssele Wind Farm Site V." https://english.rvo.nl/subsidies-financing/offshore-wind-energy/borssele-site-v (accessed 2024).

[27] Power_Technology. "Borkum Riffgrund 2 Offshore Wind Farm, North Sea." https://www.power-technology.com/projects/borkum-riffgrund-2-offshore-windfarm-north-sea/ (accessed 2024).

[28] Ørsted. "Our offshore wind farms." https://orsted.com/en/what-we-do/renewable-energy-solutions/offshore-wind/our-offshore-wind-farms (accessed 2024).

[29] 4coffshore. "Global Offshore Map." TGS. https://map.4coffshore.com/offshorewind/index.aspx?lat=53.952&lon=6.487&wfid=DE30 (accessed 2024).

[30] Energy-Charts. "Annual electricity generation from wind offshore in Germany." https://www.energy-

charts.info/charts/energy/chart.htm?l=en&c=DE&source=wind_offshore_unit_eex&interval=year&year=2023&legendItems=iy6 (accessed 2024).

[31] Ørsted. "Hornsea 2 Offshore Wind Farm." https://orsted.co.uk/energy-solutions/offshore-wind/our-wind-farms/hornsea2 (accessed 2024).

[32] Power_Technology. "Hornsea Project Two, North Sea." https://www.power-technology.com/projects/hornsea-project-two-north-sea/ (accessed 2024).

[33] THE_CROWN_ESTATE, "UK Offshore Wind Report 2023," The Crown estate, 2024. [Online]. Available: https://www.thecrownestate.co.uk/our-business/marine/offshore-wind-report-2023

[34] D. Trabucchi, L. Vollmer, and M. Kühn, "3-D shear-layer model for the simulation of multiple wind turbine wakes: description and first assessment," *Wind Energy Science*, vol. 2, no. 2, pp. 569-586, 2017, doi: 10.5194/wes-2-569-2017.

[35] P. W. F. González-Longatt, V. Terzija, "Wake effect in wind farm performance: Steady-state and dynamic behaviour," *Renewable Energy*, vol. 39, no. 1, pp. 329-338, 2011, doi: https://doi.org/10.1016/j.renene.2011.08.053.

[36] F. V. Mühle, "An ecperimental study on rotor-wake interactions of wind turbines," PhD, Norwegian University of Life Sciences, 2018. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2614550

[37] J. S. Sikkeland, "Power Prediction and Wake Losses in Offshore Wind Farms with the Dynamic Wake Meandering Model," Master, Norwegian University of Life Sciences, 2020. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2725555

[38] C. Hasager *et al.*, "Using Satellite SAR to Characterize the Wind Flow around Offshore Wind Farms," *Energies*, vol. 8, no. 6, pp. 5413-5439, 2015, doi: 10.3390/en8065413.

[39] DTU_Wind_Energy. "Welcome to PyWake." https://topfarm.pages.windenergy.dtu.dk/PyWake/ (accessed 2024).

[40] DTU_Wind_Energy. "Overview." https://topfarm.pages.windenergy.dtu.dk/PyWake/notebooks/Overview.html#The-PyWake-Philosophy (accessed 2024).

[41] DTU_Wind_Energy. "Site classes." https://topfarm.pages.windenergy.dtu.dk/PyWake/api/Site.html#xrsite (accessed 2024).

[42] DTU_Wind_Energy. "Site Object." https://topfarm.pages.windenergy.dtu.dk/PyWake/notebooks/Site.html#UniformWeibullSite (accessed 2024).

[43] S.-E. Gryning, R. Floors, A. Peña, E. Batchvarova, and B. Brümmer, "Weibull Wind-Speed Distribution Parameters Derived from a Combination of Wind-Lidar and Tall-Mast Measurements Over Land, Coastal and Marine Sites," *Boundary-Layer Meteorology*, vol. 159, no. 2, pp. 329-348, 2015, doi: 10.1007/s10546-015-0113-x.

[44] Dutch_Offshore_Wind_Atlas. "Weibull parameters a and k per wind farm zone and Cabauw." https://www.dutchoffshorewindatlas.nl/atlas/image-library/image-library/weibull-parameters (accessed April, 2024).

[45] The_Swiss_Wind_Power_Data_Website. "Weibull Calculator." https://wind-data.ch/tools/weibull.php?lng=en (accessed March, 2024).

[46] J. Frost. "Weibull Distribution: Uses, Parrameters & Examples." https://statisticsbyjim.com/probability/weibull-distribution/#comments (accessed April, 2024).

[47] T. P. Chang, "Estimation of wind energy potential using different probability density functions," *Applied Energy,* vol. 88, no. 5, pp. 1848-1856, 2010, doi: https://doi.org/10.1016/j.apenergy.2010.11.010.

[48] DTU_Wind_Energy. "Wind Farm Simulation." https://topfarm.pages.windenergy.dtu.dk/PyWake/notebooks/RunWindFarmSimulation.html (accessed 2024).

[49] P. v. d. L. Tuhfe Göcmen, Pierre-Elouan Réthoré, Alfredo peña Diaz, Gunner Chr. Larsen, Søren Ott, "Wind turbine wake models developed at the technical university of Denmark: A review," *Renewable and Sustainable Energy Reviews,* vol. 60, pp. 752-769, 2015, doi: https://doi.org/10.1016/j.rser.2016.01.113.

[50] N. O. Jensen, "A note on wind generator interaction," Risø National Laboratory, 1983. [Online]. Available: https://backend.orbit.dtu.dk/ws/portalfiles/portal/55857682/ris_m_2411.pdf

[51] DTU_Wind_Energy. "Wake Deficit Models." https://topfarm.pages.windenergy.dtu.dk/PyWake/notebooks/WakeDeficitModels.html (accessed 2024).

[52] M. al Halabi, G. Elseth, and A. Johnsen Gyllensteen, "Comparing wind farm production data to engineering wake model simulations," ed: University of South-Eastern Norway, 2023.

[53] N. N. Davis *et al.*, "The Global Wind Atlas: A High-Resolution Dataset of Climatologies and Associated Web-Based Application," (in English), *Bulletin of the American Meteorological Society,* vol. 104, no. 8, pp. E1507-E1525, 01 Aug. 2023 2023, doi: https://doi.org/10.1175/BAMS-D-21-0075.1.

[54] GWA_App. "Terms Of Use." https://globalwindatlas.info/en/about/TermsOfUse (accessed March, 2024).

[55] GWA_App. "Introduction." https://globalwindatlas.info/en/about/introduction (accessed March, 2024).

[56] GWA_App. "Frequently Asked Questions." https://globalwindatlas.info/en/about/faq (accessed march, 2024).

[57] GWA_App. "Methodology." https://globalwindatlas.info/en/about/method (accessed 2024).

[58] RapidTables. "Online pixel ruler." https://www.rapidtables.com/web/tools/pixel-ruler.html (accessed 2024).

[59] DTU_Wind_Energy. "Quickstart." https://topfarm.pages.windenergy.dtu.dk/PyWake/notebooks/Quickstart.html (accessed 2024).

Appendices

Note: if copying the code over to Python make sure indents are correct.

Appendix A Python code for Borssele III & IV

Appendix B Python code for Borssele I & II

Appendix C Python code for Borssele Wind Farm Zone

Appendix D Python code for Borkum Riffgrund II

Appendix E Python code for Hornsea Project 2

Appendix A – Borssele III & IV

```
"""
This is a model based on the Borssele III and IV offshore wind farm.
The model have been made using the already existing Hornsrev1 site
as a basis, making it compatible with the PyWake tutorial.
"""
from py_wake import np
from py_wake.site._site import UniformWeibullSite
from py_wake.wind_turbines import WindTurbine
from py_wake.wind_turbines.power_ct_functions import PowerCtTabular
import matplotlib.pyplot as plt
"""The wind farm consists of 77 wind turbines with following
coordinates:"""
wt_x = [-11731, -10385, -8962, -7654, -6231, -10615, -8962, -7192, -
5231, -4115, -2769, -1269, -9346, -7692, -6038, -3885, -8346, -6346,
-4231, -2154, 0, -7192, -5077, -3423, -1923, -5769, -3154, -4577, -
3000, -1577, -269, 808, -2846, -1577, -115, 1154, 1769, -1346, 192,
1346, 2731, -1154, 385, 1962, 3077, 385, 846, 2500, 3692, 4615,
1692, 1885, 3500, 4231, 3154, 3231, 4539, 4577, 4731, 5038, 5231,
5423, 5769, 6115, 6423, -3731, -2577, -1192, 154, 1346, 2769, 3923,
-2269, -654, 577, 1692, 2538]
wt_y = [1308, 1692, 2077, 2577, 3038, 269, 923, 1384, 2461, 2038,
1500, 1038, -1038, -615, -153, 423, -2038, -1615, -1077, -577, 0, -
3192, -3154, -2269, -2231, -4423, -3885, -5615, -5769, -3846, -2269,
-885, -7308, -5423, -3769, -2308, -1308, -7154, -5192, -3577, -1731,
-9038, -6923, -4923, -3462, -10346, -8615, -6962, -5462, -4154, -
11731, -9731, -8654, -7077, -13115, -11038, -14538, -13231, -11923,
-10462, -9154, -7808, -6231, -4769, -3346, 3885, 4192, 4731, 5154,
5462, 5885, 6308, 3308, 3385, 2154, 3577, 4692]
power_curve = np.array([[3.0, 0.0],
                        [4.0, 249.0],
                        [5.0, 613.0],
                        [6.0, 1226.0],
                        [7.0, 2030.0],
                        [8.0, 3123.0],
                        [9.0, 4444.0],
                        [10.0, 5900.0],
                        [11.0, 7299.0],
                        [12.0, 8601.0],
                        [13.0, 9272.0],
                        [14.0, 9500.0],
                        [15.0, 9500.0],
                        [16.0, 9500.0],
                        [17.0, 9500.0],
```

```
                              [18.0, 9500.0],
                              [19.0, 9500.0],
                              [20.0, 9500.0],
                              [21.0, 9500.0],
                              [22.0, 9500.0],
                              [23.0, 9500.0],
                              [24.0, 9500.0],
                              [25.0, 9500.0]]) * [1, 1000]
"""Reference Ct-curve from the DTU 10 MW turbine
As the Ct-curve of the V164-9.5 MW wind turbine wasnt accessible a
reference Ct-curve from a similar sized wind turbine already in
PyWake was used."""
ct_curve = np.array([[3, 0.0],
                          [4, 0.923],
                          [5, 0.919],
                          [6, 0.904],
                          [7, 0.858],
                          [8, 0.814],
                          [9, 0.814],
                          [10, 0.814],
                          [11, 0.814],
                          [12, 0.577],
                          [13, 0.419],
                          [14, 0.323],
                          [15, 0.259],
                          [16, 0.211],
                          [17, 0.175],
                          [18, 0.148],
                          [19, 0.126],
                          [20, 0.109],
                          [21, 0.095],
                          [22, 0.084],
                          [23, 0.074],
                          [24, 0.066],
                          [25, 0.059],
                          ])
class V164(WindTurbine):
    def __init__(self, method='linear'):
        """
        Parameters
```

```
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='V164', diameter=164,
hub_height=109,

powerCtFunction=PowerCtTabular(power_curve[:, 0], power_curve[:, 1],
'w',

ct_curve[:, 1], method=method))
"""The Weibull parameters are data collected from Global wind atlas
at the coordinatesN 51.438601°, E 3.026184° of the windfarm at 100m
and rougness 0.00."""
class Borssele3and4(UniformWeibullSite):
    def __init__(self, ti=.1, shear=None):
        f = [6, 8, 7, 7, 4, 4, 9, 13, 18, 11, 7, 6]
        """f: This is the probability of each wind direction."""
        a = [8.60, 8.89, 9.52, 9.81, 9.35, 8.90, 12.26, 12.08,
12.57, 9.96, 8.95, 9.05]
        """a: This is the Weibull scaling parameters for each wind
direction sector."""
        k = [2.213, 2.400, 2.732, 2.639, 3.014, 2.311, 2.592, 2.736,
2.482, 2.068, 1.889, 1.979]
        """k: This is the Weibull shape parameter for each wind
direction sector."""
        UniformWeibullSite.__init__(self, np.array(f) / np.sum(f),
a, k, ti=ti, shear=shear)
        self.initial_position = np.array([wt_x, wt_y]).T


def main():
    wt = V164()
    print('Turbine diameter[m]:', wt.diameter())
    print('Hub height[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
```

```
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.show()

if __name__ == '__main__':
    main()

wts=V164()
s = Borssele3and4()
x, y = wt_x, wt_y
plt.figure()
wts.plot_xy(x,y)
plt.xlim()
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.legend()

plt.figure()
Borssele3and4().plot_wd_distribution(n_wd=12);
plt.title('Wind rose')
"""
Here the AEP and wake losses of the model is calculated using the
NOJ wake deficit model.
"""
from py_wake import NOJ
windTurbines = V164()
site = Borssele3and4()
noj = NOJ(site,windTurbines)
simulationResult = noj(wt_x,wt_y)
simulationResult.aep
print (
"Total AEP of Borssele III & IV: %fGWh"
%simulationResult.aep().sum())
wf_model = NOJ(site, windTurbines)
sim_res = wf_model(x, y,
                   # wind turbine positions
                   h=None,
                   #wind turbine heights
                   type=0,
                   #Wind turbine types
                   wd=None,
```

```
                     #Wind direction
                     ws=None,
                     #Wind speed
                    )
sim_res
sim_res.aep()
aep_with_wake_loss = sim_res.aep().sum().data
aep_witout_wake_loss = sim_res.aep(with_wake_loss=False).sum().data
wake_loss = aep_witout_wake_loss - aep_with_wake_loss

print('wake loss: %f'%wake_loss, 'GWh per year')
```

Appendix B – Borssele I & II

```
"""
This is a model based on the Borssele I and II offshore wind farm.
The model have been made using the already existing Hornsrev1 site
as a basis, making it compatible with the PyWake tutorial.
"""
from py_wake import np
from py_wake.site._site import UniformWeibullSite
from py_wake.wind_turbines import WindTurbine
from py_wake.wind_turbines.power_ct_functions import PowerCtTabular
import matplotlib.pyplot as plt
"""The wind farm consists of 94 SG8.0-167 DD wind turbines with
following coordinates:"""
wt_x = [3838, 2416, 3391, 4142, 5036, 5868, 6619, 7452, 8711, 8508,
8305, 7066, 8000, 6477, 5523, 4934, 5442, 6782, 7716, 6802, 6863,
9462, 9706, 10477, 8223, 8508, 8751, 9198, 10497, 11330, 11939,
12223, 12406, 12528, 12792, 12995, 13117, 13340, 12244, 11330,
10355, 9259, 10051, 9970, 10863, 11228, 11492, 7756, 8832, 9970,
11066, 12142, 12832, 12629, 12447, 12244, 12000, 11777, 11635,
11391, 10437, 9563, 8711, 6843, 6091, 6294, 6355, 6640, 6883, 7188,
7411, 7594, 8832, 10071, 11350, 8690, 9888, 11330, 8447, 9584,
11046, 8122, 9299, 10782, 7817, 9056, 10701, 7614, 8893, 10274,
7594, 9076, 10193, 8102]
wt_y = [365, 995, 2152, 3269, 4426, 5462, 6457, 7513, 7939, 6883,
5645, 5178, 4000, 3777, 2660, 1259, -365, 1482, 2437, 183, -1117,
5178, 6416, 5340, -1665, -183, 1320, 2741, 3249, 4183, 3269, 2234,
1239, 223, -914, -1949, -2944, -4122, -3635, -3208, -2640, -2173, -
893, 690, 1929, 81, -1584, -4061, -4670, -5239, -5685, -6274, -6863,
-7959, -9137, -10254, -11411, -12589, -13665, -14802, -15939, -
16914, -17970, -16792, -15797, -14274, -12731, -11046, -9746, -8284,
-6883, -5462, -6132, -6883, -7168, -7614, -8467, -8426, -9178, -
9970, -9665, -10579, -11431, -10964, -12122, -12812, -12122, -13645,
-14335, -13543, -15310, -15756, -14822, -16853]

power_curve = np.array([[3.0, 0.0],
                        [4.0, 169.0],
                        [5.0, 593.0],
                        [6.0, 1307.0],
                        [7.0, 2186.0],
                        [8.0, 3278.0],
                        [9.0, 4687.0],
                        [10.0, 6112.0],
                        [11.0, 7249.0],
                        [12.0, 7795.0],
                        [13.0, 7947.0],
                        [14.0, 8000.0],
                        [15.0, 8000.0],
                        [16.0, 8000.0],
                        [17.0, 8000.0],
                        [18.0, 8000.0],
                        [19.0, 8000.0],
                        [20.0, 8000.0],
                        [21.0, 8000.0],
                        [22.0, 8000.0],
                        [23.0, 8000.0],
                        [24.0, 8000.0],
                        [25.0, 8000.0]]) * [1, 1000]
"""
```

```
As the Ct-curve of the SG8.0-167 DD wind turbine wasnt accessible, a
reference Ct-curve was used.
The Ct-curve used was the one of the LW 8 MW reference wind turbine,
which is a wind turbine model based on the similarly sized Vestas
V164-8 MW wind turbine.
Description of an 8 MW reference wind turbine:
https://iopscience.iop.org/article/10.1088/1742-
6596/753/9/092013#references
"""
ct_curve = np.array([[3, 0.0],
                     [4, 0.92],
                     [5, 0.85],
                     [6, 0.82],
                     [7, 0.80],
                     [8, 0.78],
                     [9, 0.76],
                     [10, 0.73],
                     [11, 0.67],
                     [12, 0.52],
                     [13, 0.39],
                     [14, 0.30],
                     [15, 0.24],
                     [16, 0.19],
                     [17, 0.16],
                     [18, 0.14],
                     [19, 0.12],
                     [20, 0.10],
                     [21, 0.09],
                     [22, 0.08],
                     [23, 0.07],
                     [24, 0.06],
                     [25, 0.05],
                     ])

class SG8_167(WindTurbine):
    def __init__(self, method='linear'):
        """
        Parameters
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='SG8_167', diameter=167,
hub_height=116,

powerCtFunction=PowerCtTabular(power_curve[:, 0], power_curve[:, 1],
'w',

ct_curve[:, 1], method=method))
"""The Weibull parameters are data collected from Global wind atlas
at the coordinates N 51.438601°, E 3.026184° of the windfarm at
100m."""
class Borssele1and2(UniformWeibullSite):
    def __init__(self, ti=.1, shear=None):
        f = [6, 8, 7, 7, 4, 4, 9, 13, 18, 11, 7, 6]
        """f: This is the probability of each wind direction."""
        a = [8.60, 8.89, 9.52, 9.81, 9.35, 8.90, 12.26, 12.08,
12.57, 9.96, 8.95, 9.05]
        """a: This is the Weibull scaling parameters for each wind
direction sector."""
```

```
        k = [2.213, 2.400, 2.732, 2.639, 3.014, 2.311, 2.592, 2.736,
2.482, 2.068, 1.889, 1.979]
        """k: This is the Weibull shape parameter for each wind
direction sector."""
        UniformWeibullSite.__init__(self, np.array(f) / np.sum(f),
a, k, ti=ti, shear=shear)
        self.initial_position = np.array([wt_x, wt_y]).T

def main():
    wt = SG8_167()
    print('SG8.0-167 DD rotor diameter[m]:', wt.diameter())
    print('Hub heigh[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.show()

if __name__ == '__main__':
    main()

wts=SG8_167()
s = Borssele1and2
x, y = wt_x, wt_y
plt.figure()
wts.plot_xy(x,y)
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.legend()

plt.figure()
Borssele1and2().plot_wd_distribution(n_wd=12);
plt.title('Wind rose')
"""
Here the AEP and wake losses of the model is calculated using the
NOJ wake deficit model.
"""
from py_wake import NOJ
windTurbines = SG8_167()
site = Borssele1and2()
noj = NOJ(site,windTurbines)
simulationResult = noj(wt_x,wt_y)
simulationResult.aep
print (
"Total AEP of Borssele I & II: %f GWh"%simulationResult.aep().sum())
wf_model = NOJ(site, windTurbines)
sim_res = wf_model(x, y,
                   # wind turbine positions
                   h=None,
                   #wind turbine heights
                   type=0,
                   #Wind turbine types
```

```
                   wd=None,
                   #Wind direction
                   ws=None,
                   #Wind speed
                  )
sim_res
sim_res.aep()
aep_with_wake_loss = sim_res.aep().sum().data
aep_witout_wake_loss = sim_res.aep(with_wake_loss=False).sum().data
wake_loss = aep_witout_wake_loss - aep_with_wake_loss
print('wake loss: %f'%wake_loss, 'GWh per year')
```

Appendix C – Borssele Wind Farm Zone

```
"""
This is a model based on the Borssele offshore wind zone.
The model have been made using the already existing Hornsrev1 site
as a basis.
"""
from py_wake import np
from py_wake.site._site import UniformWeibullSite
from py_wake.wind_turbines import WindTurbine
from py_wake.wind_turbines.power_ct_functions import PowerCtTabular
import matplotlib.pyplot as plt
"""
The Borssele wind farm zone consists of two wind farms taking up two
zones each and test site. In this model the two wind farms are
modelled.
These two wind farms are Borssele I & II and Borssele III & IV,
consisting of 94 SG8.0-167 DD and 77 V164-9.5 MW wind turbines.
The following is the coordinates of the wind turbines"""
wt_x = [-11731, -10385, -8962, -7654, -6231, -10615, -8962,
        -7192, -5231, -4115, -2769, -1269, -9346, -7692,
        -6038, -3885, -8346, -6346, -4231, -2154, 0, -7192,
        -5077, -3423, -1923, -5769, -3154, -4577, -3000,
        -1577, -269, 808, -2846, -1577, -115, 1154, 1769,
        -1346, 192, 1346, 2731, -1154, 385, 1962, 3077,
        385, 846, 2500, 3692, 4615, 1692, 1885, 3500,
        4231, 3154, 3231, 4539, 4577, 4731, 5038, 5231,
        5423, 5769, 6115, 6423, -3731, -2577, -1192, 154,
        1346, 2769, 3923, -2269, -654, 577, 1692, 2538,
        3838, 2416, 3391, 4142, 5036, 5868, 6619, 7452,
        8711, 8508, 8305, 7066, 8000, 6477, 5523, 4934,
        5442, 6782, 7716, 6802, 6863, 9462, 9706, 10477,
        8223, 8508, 8751, 9198, 10497, 11330, 11939,
        12223, 12406, 12528, 12792, 12995, 13117,
        13340, 12244, 11330, 10355, 9259, 10051, 9970,
        10863, 11228, 11492, 7756, 8832, 9970, 11066,
        12142, 12832, 12629, 12447, 12244, 12000,
        11777, 11635, 11391, 10437, 9563, 8711, 6843,
        6091, 6294, 6355, 6640, 6883, 7188, 7411, 7594,
        8832, 10071, 11350, 8690, 9888, 11330, 8447, 9584,
        11046, 8122, 9299, 10782, 7817, 9056, 10701,
        7614, 8893, 10274, 7594, 9076, 10193, 8102]
wt_y = [1308, 1692, 2077, 2577, 3038, 269, 923, 1384,
        2461, 2038, 1500, 1038, -1038, -615, -153,
        423, -2038, -1615, -1077, -577, 0, -3192,
        -3154, -2269, -2231, -4423, -3885, -5615,
        -5769, -3846, -2269, -885, -7308, -5423, -3769,
        -2308, -1308, -7154, -5192, -3577, -1731, -9038,
        -6923, -4923, -3462, -10346, -8615, -6962, -5462,
        -4154, -11731, -9731, -8654, -7077, -13115,
        -11038, -14538, -13231, -11923, -10462, -9154,
        -7808, -6231, -4769, -3346, 3885, 4192, 4731,
        5154, 5462, 5885, 6308, 3308, 3385, 2154, 3577,
```

```
        4692, 365, 995, 2152, 3269, 4426, 5462, 6457,
        7513, 7939, 6883, 5645, 5178, 4000, 3777, 2660,
        1259, -365, 1482, 2437, 183, -1117, 5178, 6416,
        5340, -1665, -183, 1320, 2741, 3249, 4183,
        3269, 2234, 1239, 223, -914, -1949, -2944,
        -4122, -3635, -3208, -2640, -2173, -893, 690,
        1929, 81, -1584, -4061, -4670, -5239, -5685,
        -6274, -6863, -7959, -9137, -10254, -11411,
        -12589, -13665, -14802, -15939, -16914,
        -17970, -16792, -15797, -14274, -12731,
        -11046, -9746, -8284, -6883, -5462, -6132,
        -6883, -7168, -7614, -8467, -8426, -9178,
        -9970, -9665, -10579, -11431, -10964,
        -12122, -12812, -12122, -13645, -14335,
        -13543, -15310, -15756, -14822, -16853]
"""This is the power curve of the V164-9.5 MW wind turbine"""
power_curve1 = np.array([[3.0, 0.0],
                        [4.0, 169.0],
                        [5.0, 593.0],
                        [6.0, 1307.0],
                        [7.0, 2186.0],
                        [8.0, 3278.0],
                        [9.0, 4687.0],
                        [10.0, 6112.0],
                        [11.0, 7249.0],
                        [12.0, 7795.0],
                        [13.0, 7947.0],
                        [14.0, 8000.0],
                        [15.0, 8000.0],
                        [16.0, 8000.0],
                        [17.0, 8000.0],
                        [18.0, 8000.0],
                        [19.0, 8000.0],
                        [20.0, 8000.0],
                        [21.0, 8000.0],
                        [22.0, 8000.0],
                        [23.0, 8000.0],
                        [24.0, 8000.0],
                        [25.0, 8000.0]]) * [1, 1000]
"""Reference Ct-curve from the DTU 10 MW turbine
As the Ct-curve of the V164-9.5 MW wind turbine wasnt accessible a
reference Ct-curve from a similar sized wind turbine already in
PyWake was used."""
ct_curve1 = np.array([[3, 0.0],
                     [4, 0.92],
```

```
                     [5, 0.85],
                     [6, 0.82],
                     [7, 0.80],
                     [8, 0.78],
                     [9, 0.76],
                     [10, 0.73],
                     [11, 0.67],
                     [12, 0.52],
                     [13, 0.39],
                     [14, 0.30],
                     [15, 0.24],
                     [16, 0.19],
                     [17, 0.16],
                     [18, 0.14],
                     [19, 0.12],
                     [20, 0.10],
                     [21, 0.09],
                     [22, 0.08],
                     [23, 0.07],
                     [24, 0.06],
                     [25, 0.05],
                     ])
"""This is the power curve of the SG8.0-167 DD wind turbine"""
power_curve2 = np.array([[3.0, 0.0],
                        [4.0, 249.0],
                        [5.0, 613.0],
                        [6.0, 1226.0],
                        [7.0, 2030.0],
                        [8.0, 3123.0],
                        [9.0, 4444.0],
                        [10.0, 5900.0],
                        [11.0, 7299.0],
                        [12.0, 8601.0],
                        [13.0, 9272.0],
                        [14.0, 9500.0],
                        [15.0, 9500.0],
                        [16.0, 9500.0],
                        [17.0, 9500.0],
                        [18.0, 9500.0],
```

```
                                  [19.0, 9500.0],
                                  [20.0, 9500.0],
                                  [21.0, 9500.0],
                                  [22.0, 9500.0],
                                  [23.0, 9500.0],
                                  [24.0, 9500.0],
                                  [25.0, 9500.0]]) * [1, 1000]
"""As the Ct-curve of the SG8.0-167 DD wind turbine wasnt accessible
a reference Ct-curve was used.

The Ct-curve used was the one of the LW 8 MW reference wind turbine,
which is a wind turbine model based on the Vestas V164-8 MW wind
turbine.

Description of an 8 MW reference wind turbine:
https://iopscience.iop.org/article/10.1088/1742-
6596/753/9/092013#references"""

ct_curve2 = np.array([[3, 0.0],
                      [4, 0.923],
                      [5, 0.919],
                      [6, 0.904],
                      [7, 0.858],
                      [8, 0.814],
                      [9, 0.814],
                      [10, 0.814],
                      [11, 0.814],
                      [12, 0.577],
                      [13, 0.419],
                      [14, 0.323],
                      [15, 0.259],
                      [16, 0.211],
                      [17, 0.175],
                      [18, 0.148],
                      [19, 0.126],
                      [20, 0.109],
                      [21, 0.095],
                      [22, 0.084],
                      [23, 0.074],
                      [24, 0.066],
                      [25, 0.059],
                      ])
class SG8_167(WindTurbine):
    def __init__(self, method='linear'):
```

```
        """
        Parameters
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='SG8_167', diameter=167,
hub_height=116,

powerCtFunction=PowerCtTabular(power_curve1[:, 0], power_curve1[:,
1], 'w',

ct_curve1[:, 1], method=method))
class V164(WindTurbine):
    def __init__(self, method='linear'):
        """
        Parameters
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='V164', diameter=164,
hub_height=109,

powerCtFunction=PowerCtTabular(power_curve2[:, 0], power_curve2[:,
1], 'w',

ct_curve2[:, 1], method=method))
"""The Weibull parameters are data collected from Global wind atlas
at the coordinates N 51.438601°, E 3.026184° of the windfarm at 100m
and roughness 0.00."""
class BorsseleWfz(UniformWeibullSite):
    def __init__(self, ti=.1, shear=None):
        f = [6, 8, 7, 7, 4, 4, 9, 13, 18, 11, 7, 6]
        """f: This is the probability of each wind direction."""
        a = [8.60, 8.89, 9.52, 9.81, 9.35, 8.90, 12.26, 12.08,
12.57, 9.96, 8.95, 9.05]
        """a: This is the Weibull scaling parameters for each wind
direction sector."""
        k = [2.213, 2.400, 2.732, 2.639, 3.014, 2.311, 2.592, 2.736,
2.482, 2.068, 1.889, 1.979]
        """k: This is the Weibull shape parameter for each wind
direction sector."""
```

```
        UniformWeibullSite.__init__(self, np.array(f) / np.sum(f),
a, k, ti=ti, shear=shear)
        self.initial_position = np.array([wt_x[77:171],
wt_y[77:171]]).T


def main():
    wt = SG8_167()
    print('SG8.0-167 rotor diameter[m]:', wt.diameter())
    print('SG8.0-167 hub height[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.title('SG8.0-167 DD')
    plt.show()

if __name__ == '__main__':
    main()

def main():
    wt = V164()
    print('V164 rotor diameter[m]:', wt.diameter())
    print('V164 hub height[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
```

```
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.title('V164-9.5 MW')
    plt.show()

if __name__ == '__main__':
    main()

wts=SG8_167()
wts2=V164()
s = BorsseleWfz()
x, y = wt_x[77:171], wt_y[77:171]
x2, y2 = wt_x[0:76], wt_y[0:76]
plt.figure()
wts.plot_xy(x,y)
wts2.plot_xy(x2,y2)
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.legend()

plt.figure()
BorsseleWfz().plot_wd_distribution(n_wd=12);
plt.title('Wind rose')

"""
Here the AEP and wake losses of the model is calculated using the
NOJ wake deficit model.
"""
from py_wake import NOJ
windTurbines1 = SG8_167()
site1 = BorsseleWfz()
noj1 = NOJ(site1,windTurbines1)
windTurbines2 = V164()
site2 = BorsseleWfz()
noj2 = NOJ(site2,windTurbines2)
noj = noj2 and noj1
simulationResult = noj(wt_x,wt_y)
print ("Total AEP of the Borssele wind farm zone: %f GWh"
```

```
       %simulationResult.aep().sum())
wf_model = noj
x, y = wt_x, wt_y
sim_res = wf_model(x,y)
sim_res
sim_res.aep()
aep_with_wake_loss = sim_res.aep().sum().data
aep_witout_wake_loss = sim_res.aep(with_wake_loss=False).sum().data
wake_loss = aep_witout_wake_loss - aep_with_wake_loss
print('wake loss: %f'%wake_loss, 'GWh per year')
```

Appendix D – Borkum Riffgrund II

```
"""
This is a model based on the Borkum Riffgrund II offshore wind farm.
The model have been made using the already existing Hornsrev1 site
as a basis, making it compatible with the PyWake tutorial.
"""
from py_wake import np
from py_wake.site._site import UniformWeibullSite
from py_wake.wind_turbines import WindTurbine
from py_wake.wind_turbines.power_ct_functions import PowerCtTabular
import matplotlib.pyplot as plt
"""The wind farm consists of 56 V164-8 MW wind turbines with
following coordinates:"""
wt_x = [-5781, -4534, -3795, -2356, -5753, -5027, -3493, -2356, -
5726, -5027, -2356, -5685, -5000, -4247, -3479, -1932, -5219, -2712,
-1932, -4685, -3726, 0, -4301, -3014, -1151, 384, -3479, -2712, -
1151, 384, 1945, -2836, -1945, -1164, 384, 2110, 3027, -2274, -1164,
384, -1808, -1329, -493, 384, 1137, 1890, 2630, 3384, 4137, 4890,
5644, 6425, 5767, 7301, 7301, 7301]
wt_y = [3877, 4301, 4301, 4301, 3110, 3110, 3082, 3082, 2329, 2329,
2356, 1575, 1575, 1575, 1575, 1575, 822, 822, 890, -27, 41, 0, -507,
-740, -753, -753, -1534, -1534, -1548, -1562, -1534, -2342, - 2342,
-2342, -2342, -2507, -2384, -3055, -3123, -3123, -3644, -4260, -
4123, -3904, -3726, -3548, -3356, -3205, -3014, -2822, -2644, -2466,
3904, 3904, 3151, 2370]

power_curve = np.array([[3.0, 0.0],
                        [4.0, 100.0],
                        [5.0, 650.0],
                        [6.0, 1150.0],
                        [7.0, 1850.0],
                        [8.0, 2900.0],
                        [9.0, 4150.0],
                        [10.0, 5600.0],
                        [11.0, 7100.0],
                        [12.0, 7800.0],
                        [13.0, 8000.0],
                        [14.0, 8000.0],
                        [15.0, 8000.0],
                        [16.0, 8000.0],
                        [17.0, 8000.0],
                        [18.0, 8000.0],
                        [19.0, 8000.0],
                        [20.0, 8000.0],
                        [21.0, 8000.0],
                        [22.0, 8000.0],
                        [23.0, 8000.0],
                        [24.0, 8000.0],
                        [25.0, 8000.0]]) * [1, 1000]
"""As the Ct-curve of the V164-8 MW wind turbine wasnt accessible a
reference Ct-curve was used.
The Ct-curve used was the one of the LW 8 MW reference wind turbine,
which is a wind turbine model based on the Vestas V164-8 MW wind
turbine.
Description of an 8 MW reference wind turbine:
https://iopscience.iop.org/article/10.1088/1742-
6596/753/9/092013#references"""
ct_curve = np.array([[3, 0.0],
                     [4, 0.92],
```

```
                    [5, 0.85],
                    [6, 0.82],
                    [7, 0.80],
                    [8, 0.78],
                    [9, 0.76],
                    [10, 0.73],
                    [11, 0.67],
                    [12, 0.52],
                    [13, 0.39],
                    [14, 0.30],
                    [15, 0.24],
                    [16, 0.19],
                    [17, 0.16],
                    [18, 0.14],
                    [19, 0.12],
                    [20, 0.10],
                    [21, 0.09],
                    [22, 0.08],
                    [23, 0.07],
                    [24, 0.06],
                    [25, 0.05],
                    ])

class V164(WindTurbine):
    def __init__(self, method='linear'):
        """
        Parameters
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='V164', diameter=164,
hub_height=105,

powerCtFunction=PowerCtTabular(power_curve[:, 0], power_curve[:, 1],
'w',

ct_curve[:, 1], method=method))
"""The Weibull parameters are based on data from Global wind atlas
at the coordinates N 54.580061°, E 6.294479°of the windfarm and
height 100m and roughness 0.00."""
class BorkumRiffgrund2(UniformWeibullSite):
    def __init__(self, ti=.1, shear=None):
        f = [ 5, 4, 6, 8, 6, 6, 8, 12, 14, 12, 10, 9]
        """f: This is the probability of each wind direction."""
        a = [8.48, 7.56, 9.29, 11.10, 11.54, 10.99, 11.28, 12.67,
12.67, 11.96, 11.54, 10.26]
        """a: This is the Weibull scaling parameters for each wind
direction sector."""
        k = [2.166, 2.162, 2.178, 2.299, 2.604, 2.607, 2.002, 2.467,
2.396, 2.342, 2.568, 2.186]
        """k: This is the Weibull shape parameter for each wind
direction sector."""
        UniformWeibullSite.__init__(self, np.array(f) / np.sum(f),
a, k, ti=ti, shear=shear)
        self.initial_position = np.array([wt_x, wt_y]).T

def main():
    wt = V164()
    print('Turbine diameter[m]:', wt.diameter())
```

```
    print('Hub height[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.show()

if __name__ == '__main__':
    main()

wts=V164()
s = BorkumRiffgrund2
x, y = wt_x, wt_y
plt.figure()
wts.plot_xy(x,y)
plt.xlim()
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.legend()

plt.figure()
BorkumRiffgrund2().plot_wd_distribution(n_wd=12);
plt.title('Wind rose')

"""
Here the AEP and wake losses of the model is calculated using the
NOJ wake deficit model.
"""
from py_wake import NOJ
windTurbines = V164()
site = BorkumRiffgrund2()
noj = NOJ(site,windTurbines)
simulationResult = noj(wt_x,wt_y)
simulationResult.aep
print ("Total AEP of Borkum Riffgrund II: %f GWh"

       %simulationResult.aep().sum())
wf_model = NOJ(site, windTurbines)
sim_res = wf_model(x, y,
                   # wind turbine positions
                   h=None,
                   #wind turbine heights
                   type=0,
                   #Wind turbine types
                   wd=None,
                   #Wind direction
                   ws=None,
                   #Wind speed
                  )
sim_res
sim_res.aep()
aep_with_wake_loss = sim_res.aep().sum().data
aep_witout_wake_loss = sim_res.aep(with_wake_loss=False).sum().data
```

```
wake_loss = aep_witout_wake_loss - aep_with_wake_loss
print('wake loss: %f'%wake_loss, 'GWh per year')
```

Appendix E – Hornsea Project 2

```
"""
This is a model based on the Hornsea Project 2 offshore wind farm.
The model have been made using the already existing Hornsrev1 site
as a basis, making it compatible with the PyWake tutorial.
"""

from py_wake import np
from py_wake.site._site import UniformWeibullSite
from py_wake.wind_turbines import WindTurbine
from py_wake.wind_turbines.power_ct_functions import PowerCtTabular
import matplotlib.pyplot as plt
"""The wind farm consists of 165 SG8.0-167 DD wind turbines with
following coordinates."""
wt_x = [-600,    1240, 2680, -80,  1840, 3160, -400, 960,  2680, 4120,
        120,  3520, 5080, 640,  -160, 1320, 2840, 5160, 7000, 560,  2200,
        3880, 80,   1960, 4920, 480,  43080,     42520,     40760,
        40600,     39720,     39000,     38800,     38000,     37240,
        37240,     36360,     35080,     37400,     36800,     36160,
        35360,     34680,     33360,     32520,     34360,     33680,
        32800,     31960,     31200,     31240,     29880,     31400,
        30800,     30200,     29200,     28640,     28320,     26880,
        27440,     26760,     25960,     25160,     25280,     23840,
        24960,     24040,     23280,     22480,     21800,     21520,
        20280,     18880,     18400,     17200,     16920,     15160,
        15320,     16480,     15480,     15160,     10880,     11960,
        11600,     12240,     11640,     12480,     11680,     12680,
        12200,     11600,     13000,     12480,     12880,     13480,
        12680,     13360,     12200,     9400, 10200,     10760,
        8880, 9160, 8560, 9040, 10240,     9240, 9520, 10760,     9880,
        11000,     11840,     10760,     11680,     9760, 10560,
        9000, 7720, 6240, 7040, 6200, 7160, 5320, 6000, 6800, 7440, 8280,
        8040, 9040, 9960, 8040, 8880, 7040, 7960, 6000, 7440, 5120, 6920,
        4720, 4680, 5400, 4000, 3120, 2080, 1040, 0,    2160, -80,  760,
        2760, 3880, -40,  120,  2760, 4120, 720,  3080, 160,  5600, 840,
        120,  1240, 3000, 4560, 5920]
wt_y = [19680,   19560,     19480,     18960,     18680,     18760,
        17080,     17320,     17400,     17280,     16400,     16120,
        15840,     15480,     14360,     14480,     14360,     13640,
        13000,     13160,     13160,     12800,     11600,     11160,
        11320,     10920,     14680,     15800,     15840,     14400,
        15680,     16360,     12760,     13960,     15120,     16600,
        16520,     16840,     10400,     11400,     12360,     13680,
        14720,     16800,     17080,     10880,     11760,     13240,
        14640,     15760,     17120,     17200,     11040,     11920,
        12840,     14400,     15280,     17200,     17200,     12720,
        13800,     15080,     16240,     17360,     17680,     12160,
        13640,     14840,     16040,     17080,     18280,     18520,
        17160,     18840,     19040,     16720,     17480,     19080,
        12480,     14040,     12120,     -200, 440,  1160, 2840, 3600,
        4600, 5720, 7200, 7920, 8920, 10040,     10880,     13400,
        14760,     16000,     17360,     19040,     -160, 1000, 2320,
        560,  2560, 3440, 5000, 5920, 7440, 9080, 10280,     11600,
        13080,     14880,     16520,     17600,     18160,     19280,
        19280,     -120, -240, 920,  2240, 3160, 3680, 5000, 6160, 7480,
        8920, 11400,     12880,     14800,     14480,     16320,
        15920,     17720,     17480,     18600,     18680,     19360,
        19360,     -160, 1120, 920,  -200, 0,    -200, 0,    1480, 1640,
```

```
    2000, 2920, 3400, 3240, 4440, 5160, 5480, 5960, 7080, 6800, 8000,
    8080, 9080, 9880, 9640, 9560, 9720]

power_curve = np.array([[3.0, 0.0],
                        [4.0, 169.0],
                        [5.0, 593.0],
                        [6.0, 1307.0],
                        [7.0, 2186.0],
                        [8.0, 3278.0],
                        [9.0, 4687.0],
                        [10.0, 6112.0],
                        [11.0, 7249.0],
                        [12.0, 7795.0],
                        [13.0, 7947.0],
                        [14.0, 8000.0],
                        [15.0, 8000.0],
                        [16.0, 8000.0],
                        [17.0, 8000.0],
                        [18.0, 8000.0],
                        [19.0, 8000.0],
                        [20.0, 8000.0],
                        [21.0, 8000.0],
                        [22.0, 8000.0],
                        [23.0, 8000.0],
                        [24.0, 8000.0],
                        [25.0, 8000.0]]) * [1, 1000]

"""As the Ct-curve of the SG8.0-167 DD wind turbine wasnt
accessible, a reference Ct-curve was used.
The Ct-curve used was the one of the LW 8 MW reference wind turbine,
which is a wind turbine model based on the similarly sized Vestas
V164-8 MW wind turbine.
Description of an 8 MW reference wind turbine:
https://iopscience.iop.org/article/10.1088/1742-
6596/753/9/092013#references"""
ct_curve = np.array([[3, 0.0],
                     [4, 0.92],
                     [5, 0.85],
                     [6, 0.82],
                     [7, 0.80],
                     [8, 0.78],
                     [9, 0.76],
                     [10, 0.73],
                     [11, 0.67],
                     [12, 0.52],
                     [13, 0.39],
                     [14, 0.30],
                     [15, 0.24],
                     [16, 0.19],
                     [17, 0.16],
                     [18, 0.14],
                     [19, 0.12],
                     [20, 0.10],
                     [21, 0.09],
                     [22, 0.08],
                     [23, 0.07],
                     [24, 0.06],
                     [25, 0.05],
                     ])

class SG8_167(WindTurbine):
```

```
    def __init__(self, method='linear'):
        """
        Parameters
        ----------
        method : {'linear', 'pchip'}
            linear(fast) or pchip(smooth and gradient friendly)
interpolation
        """
        WindTurbine.__init__(self, name='SG8_167', diameter=167,
hub_height=116,

powerCtFunction=PowerCtTabular(power_curve[:, 0], power_curve[:, 1],
'w',

ct_curve[:, 1], method=method))

"""The Weibull parameters are data collected from Global wind atlas
at the coordinatesN 53.530064°, E 1.472992° of the windfarm at a
height of 100m and roughness 0.00."""
class Hornsea2(UniformWeibullSite):
    def __init__(self, ti=.1, shear=None):
        f = [6, 4, 5, 7, 5, 8, 10, 13, 16, 11, 8, 7]
        """f: This is the probability of each wind direction."""
        a = [9.74, 8.58, 9.03, 10.06, 9.08, 10.50, 11.65, 13.18,
13.08, 11.95, 10.04, 10.25]
        """a: This is the Weibull scaling parameters for each wind
direction sector."""
        k = [2.557, 2.279, 2.607, 2.232, 2.037, 2.506, 2.068, 2.428,
2.760, 2.256, 2.471, 2.182]
        """k: This is the Weibull shape parameter for each wind
direction sector."""
        UniformWeibullSite.__init__(self, np.array(f) / np.sum(f),
a, k, ti=ti, shear=shear)
        self.initial_position = np.array([wt_x, wt_y]).T

def main():
    wt = SG8_167()
    print('SG8.0-167 DD rotor diameter[m]:', wt.diameter())
    print('Hub height[m]:', wt.hub_height())

    plt.figure()
    ws = np.linspace(3, 20, 100)
    plt.plot(ws, wt.power(ws) * 1e-3, label='Power')
    c = plt.plot([], [], label='Ct')[0].get_color()
    plt.ylabel('Power [kW]')
    plt.xlabel('Wind speed [m/s]')
    ax = plt.gca().twinx()
    ax.plot(ws, wt.ct(ws), color=c)
    ax.set_ylabel('Ct')
    plt.xlabel('Wind speed [m/s]')
    plt.gcf().axes[0].legend(loc=1)
    plt.show()

if __name__ == '__main__':
    main()

wts=SG8_167()
s = Hornsea2
x, y = wt_x, wt_y
plt.figure()
wts.plot_xy(x,y)
```

```
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.legend()

plt.figure()
Hornsea2().plot_wd_distribution(n_wd=12);
plt.title('Wind rose')


"""
Here the AEP and wake losses of the model is calculated using the
NOJ wake deficit model.
"""
from py_wake import NOJ
windTurbines = SG8_167()
site = Hornsea2()
noj = NOJ(site,windTurbines)
simulationResult = noj(wt_x,wt_y)
simulationResult.aep
print ("Total AEP of Hornsea Project 2: %f GWh"
       %simulationResult.aep().sum())
wf_model = NOJ(site, windTurbines)
sim_res = wf_model(x, y,
                   # wind turbine positions
                   h=None,
                   #wind turbine heights
                   type=0,
                   #Wind turbine types
                   wd=None,
                   #Wind direction
                   ws=None,
                   #Wind speed
                  )
sim_res
sim_res.aep()
aep_with_wake_loss = sim_res.aep().sum().data
aep_witout_wake_loss = sim_res.aep(with_wake_loss=False).sum().data
wake_loss = aep_witout_wake_loss - aep_with_wake_loss
print('wake loss: %f'%wake_loss, 'GWh per year')
```