

# Wind farm flow modeling and optimization in PyWake/TOPFARM

Rafael Valotta Rodrigues

Systems Engineering and Optimization

Wind Energy Systems Division

Department of Wind and Energy Systems

# Introduction



Rafael Valotta Rodrigues

- **DTU Wind - Technical University of Denmark**

Riso Campus, Roskilde

- **Email**

ravaro@dtu.dk

rafael.rodrigues@du.edu

rafavalotta@gmail.com (personal)

- **Orcid:**

<https://orcid.org/0000-0002-2395-9957>

- **Research Gate:**

[https://www.researchgate.net/profile/Rafael\\_Rodrigues14](https://www.researchgate.net/profile/Rafael_Rodrigues14)

- **Linkedin:**

<https://www.linkedin.com/in/rafael-valotta-rodrigues-73b7628a/>

- **Twiter:**

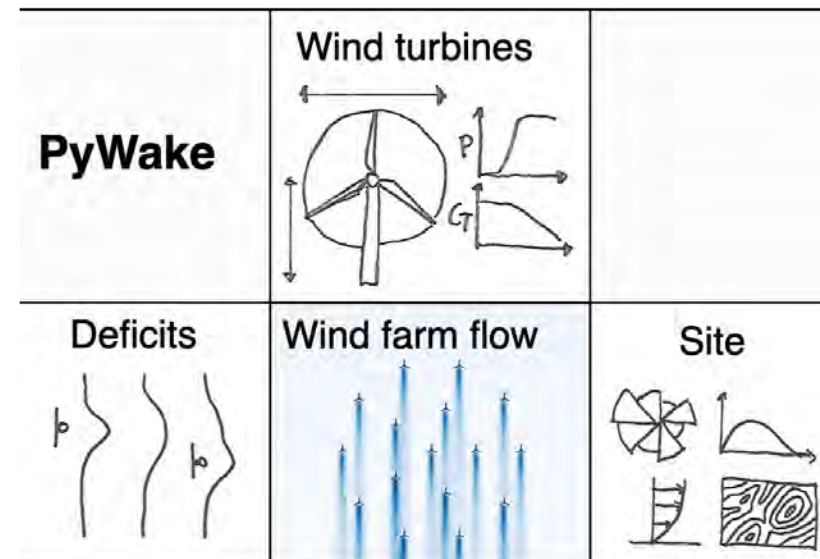
<https://twitter.com/rafavalotta>

# Agenda

- PyWake Overview
- TOPFARM Overview
- Example
- Q&A

# Wind Farm Flow Physics in PyWake

- **PyWake**: open-source tool for wind farm flow physics



```
git clone https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake.git
cd PyWake
pip install -e .[test]
```

# Wind Farm Flow Physics in PyWake



## PyWake

Pierre R  thor   (Senior Researcher)

Mads Pedersen (Development Engineer)

## The PyWake Team

PyWake is a cross-section effort at DTU

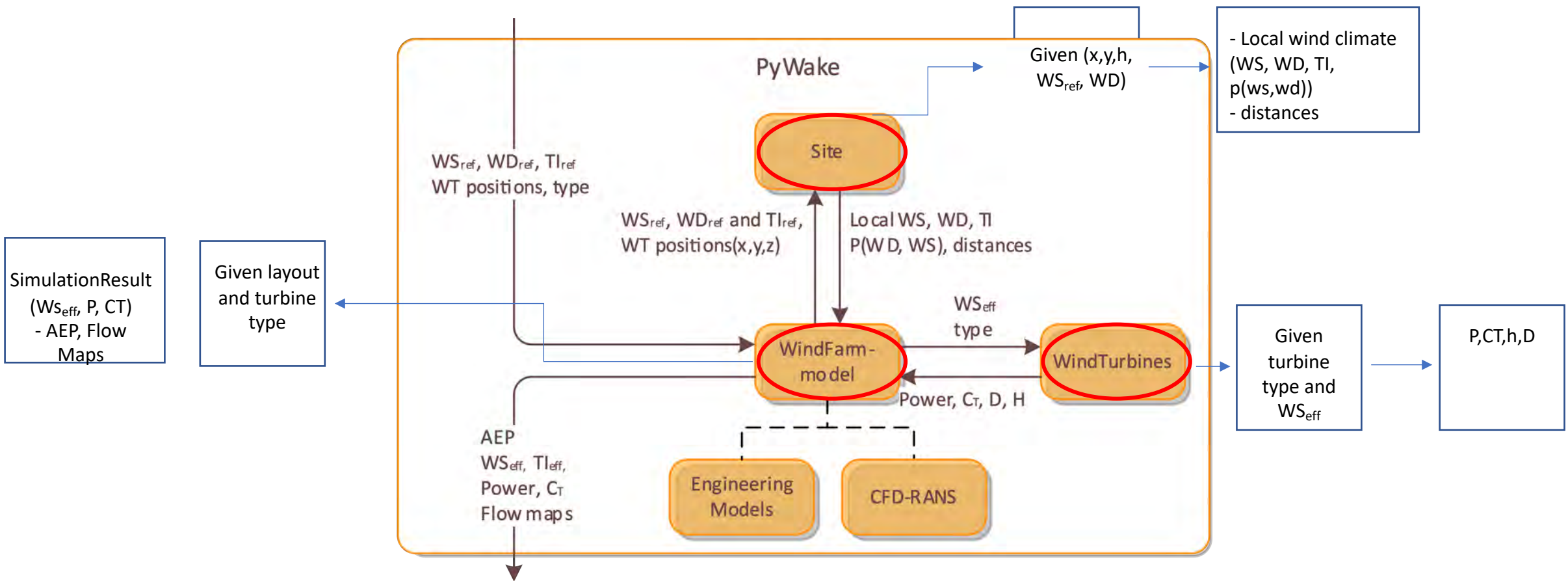


## TOPFARM

Mikkel Friis M  ller  
(Development Engineer)

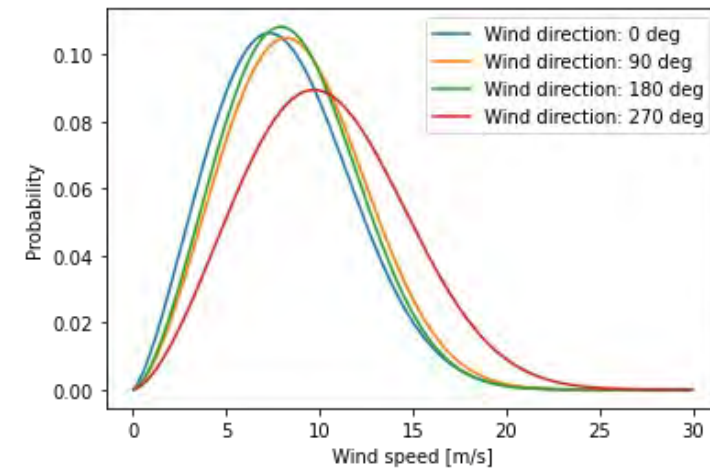
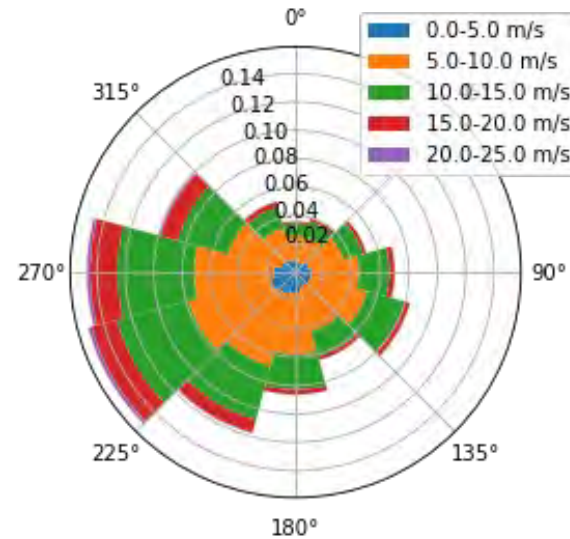
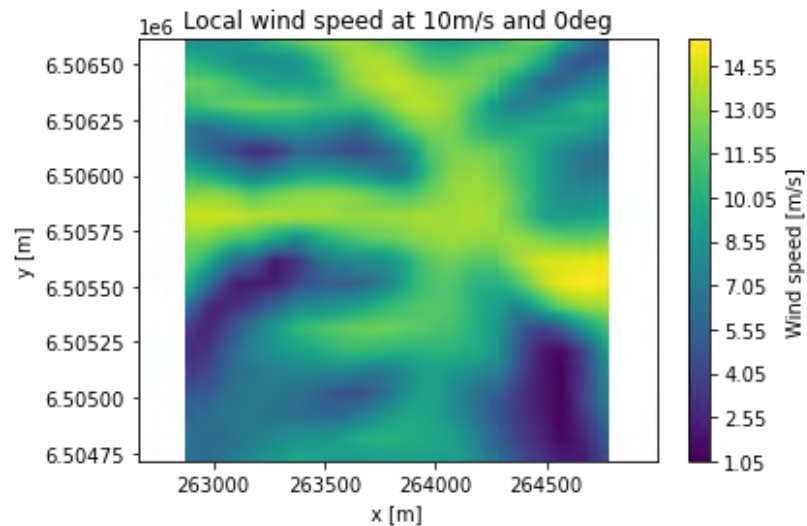


# PyWake Overview



# PyWake: Site

- Predefined Uniform Weibull Sites: Horns Rev I, Lillgrund, IEA 37
- WasP Grided Site
- XRSite



# PyWake: Site











```
from py_wake.site import WaspGridSite
from py_wake.examples.data.ParqueFicticio import ParqueFicticio_path
site = WaspGridSite.from_wasp_grd(ParqueFicticio_path)
```

site.ds





















xarray.Dataset

► Dimensions: (x: 20, y: 20, h: 2, wd: 13, ws: 23)

▼ Coordinates:

<b>x</b>	(x)	float64	2.629e+05 2.63e+05 ... 2.648e+05		
<b>y</b>	(y)	float64	6.505e+06 6.505e+06 ... 6.507e+06		
<b>h</b>	(h)	float64	30.0 200.0		
<b>wd</b>	(wd)	float64	0.0 30.0 60.0 ... 300.0 330.0 360.0		
<b>ws</b>	(ws)	int64	3 4 5 6 7 8 9 ... 20 21 22 23 24 25		

▼ Data variables:










flow_inc	(x, y, h, wd)	float64	-5.262 -8.859 ... -2.781 0.5771		
ws_mean	(x, y, h, wd)	float64	2.03 1.448 1.18 ... 6.388 5.096		
orog_spd	(x, y, h, wd)	float64	0.6241 0.4884 ... 1.031 0.9372		
Turning	(x, y, h, wd)	float64	-9.769 -2.692 ... -5.005 -4.226		
Sector_frequency	(x, y, h, wd)	float64	0.05179 0.02514 ... 0.08374 0.05213		
ti15ms	(x, y, h, wd)	float64	0.3018 0.3735 ... 0.1168 0.1403		
Weibull_A	(x, y, h, wd)	float64	2.28 1.635 1.328 ... 7.188 5.752		
Weibull_k	(x, y, h, wd)	float64	1.764 2.107 2.643 ... 1.826 2.053		
Elevation	(x, y)	float64	200.1 211.4 216.8 ... 335.2 334.9		
Speedup	(x, y, h, wd)	float64	0.6241 0.4884 ... 1.031 0.9372		

```
site.ds.isel(x=1, y=1, wd=10, h=1, ws=10)
```

xarray.Dataset

► Dimensions:

▼ Coordinates:

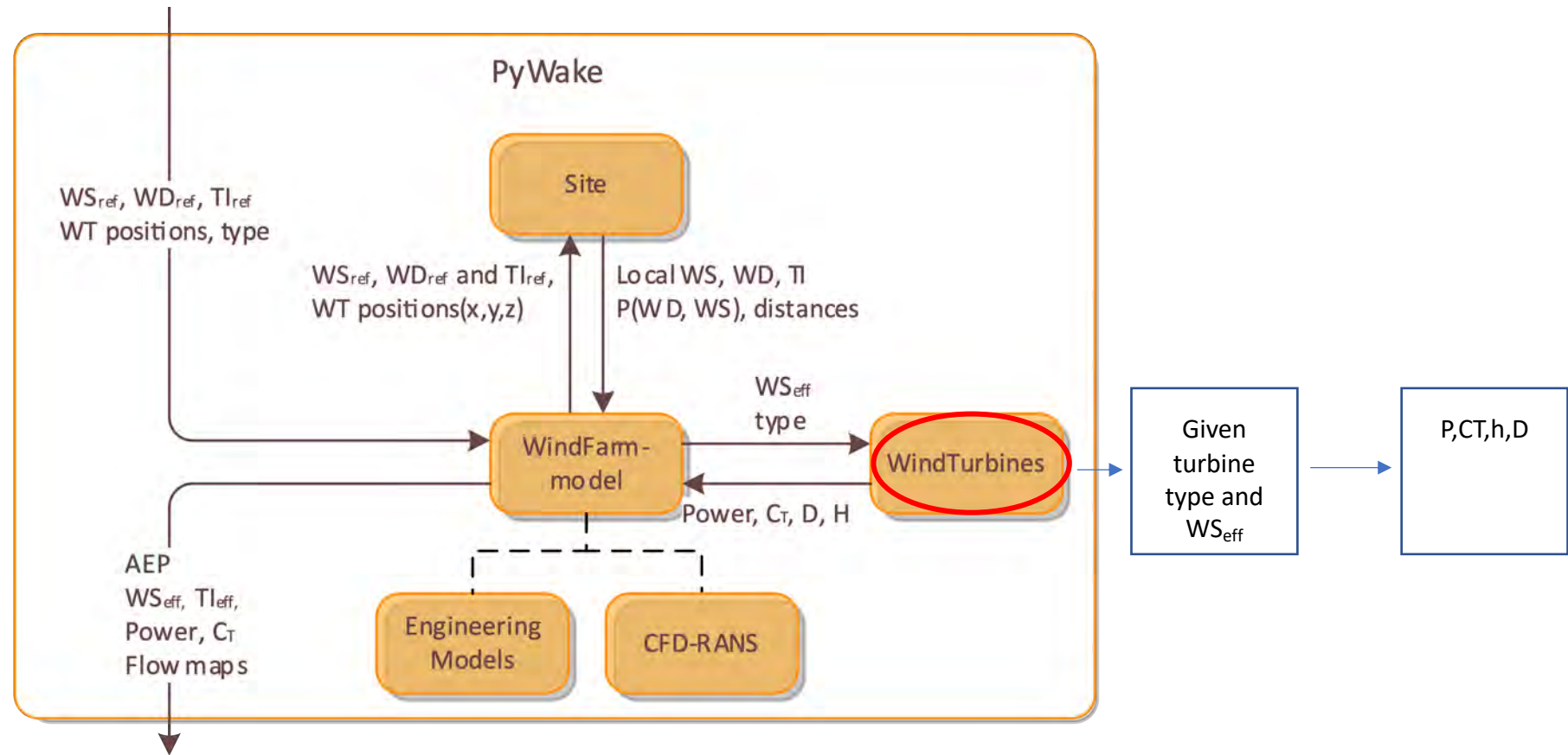
<b>x</b>	0	float64	2.63e+05		
			array(262978.)		
<b>y</b>	0	float64	6.505e+06		
			array(6504814.)		
<b>h</b>	0	float64	200.0		
			array(200.)		
<b>wd</b>	0	float64	300.0		
			array(300.)		
<b>ws</b>	0	int64	13		
			array(13)		

▼ Data variables:

flow_inc	0	float64	4.815		
			array(4.815254)		
ws_mean	0	float64	9.457		
			array(9.456785)		
orog_spd	0	float64	1.033		
			array(1.033075)		
Turning	0	float64	1.505		
			array(1.505159)		

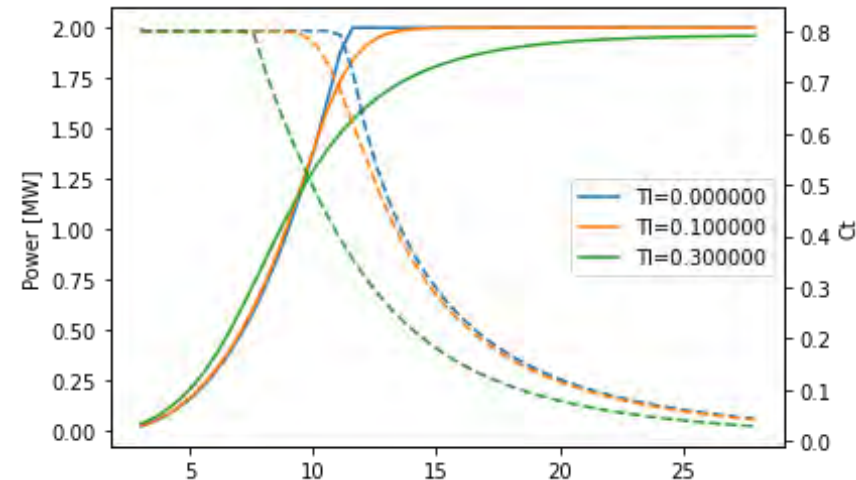
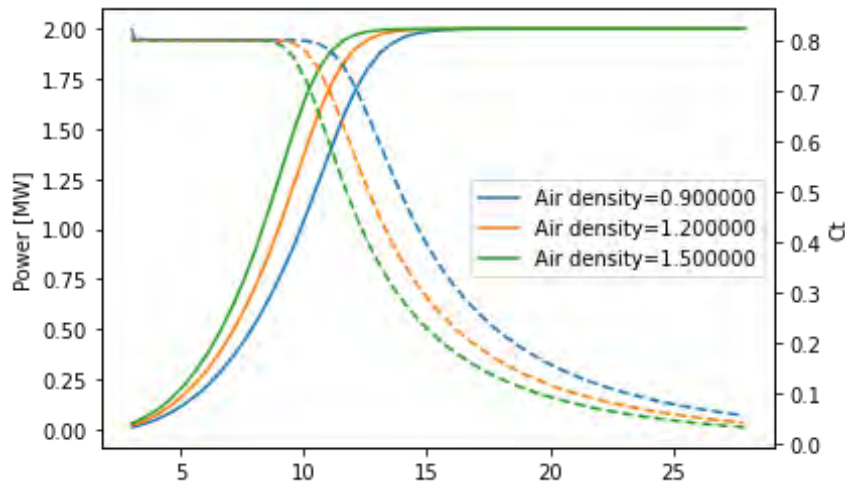
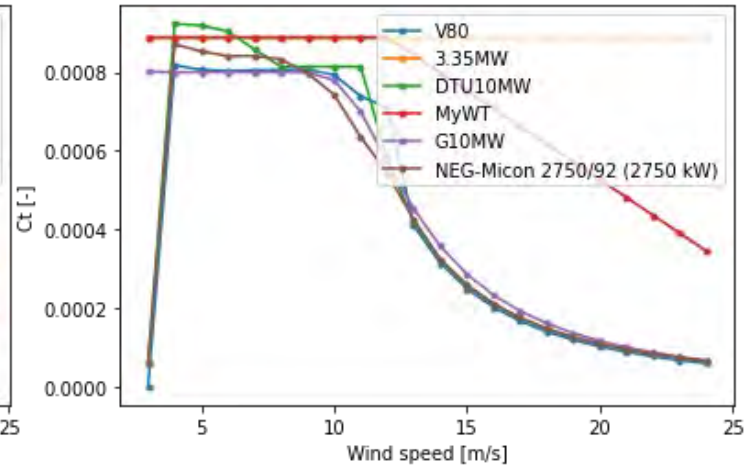
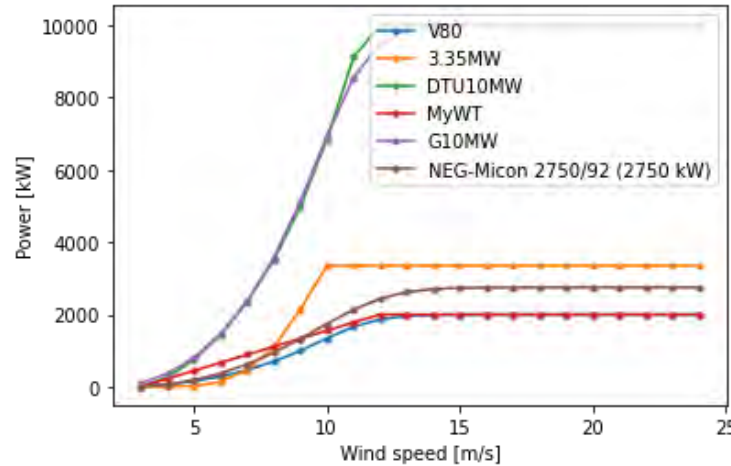


# PyWake Overview

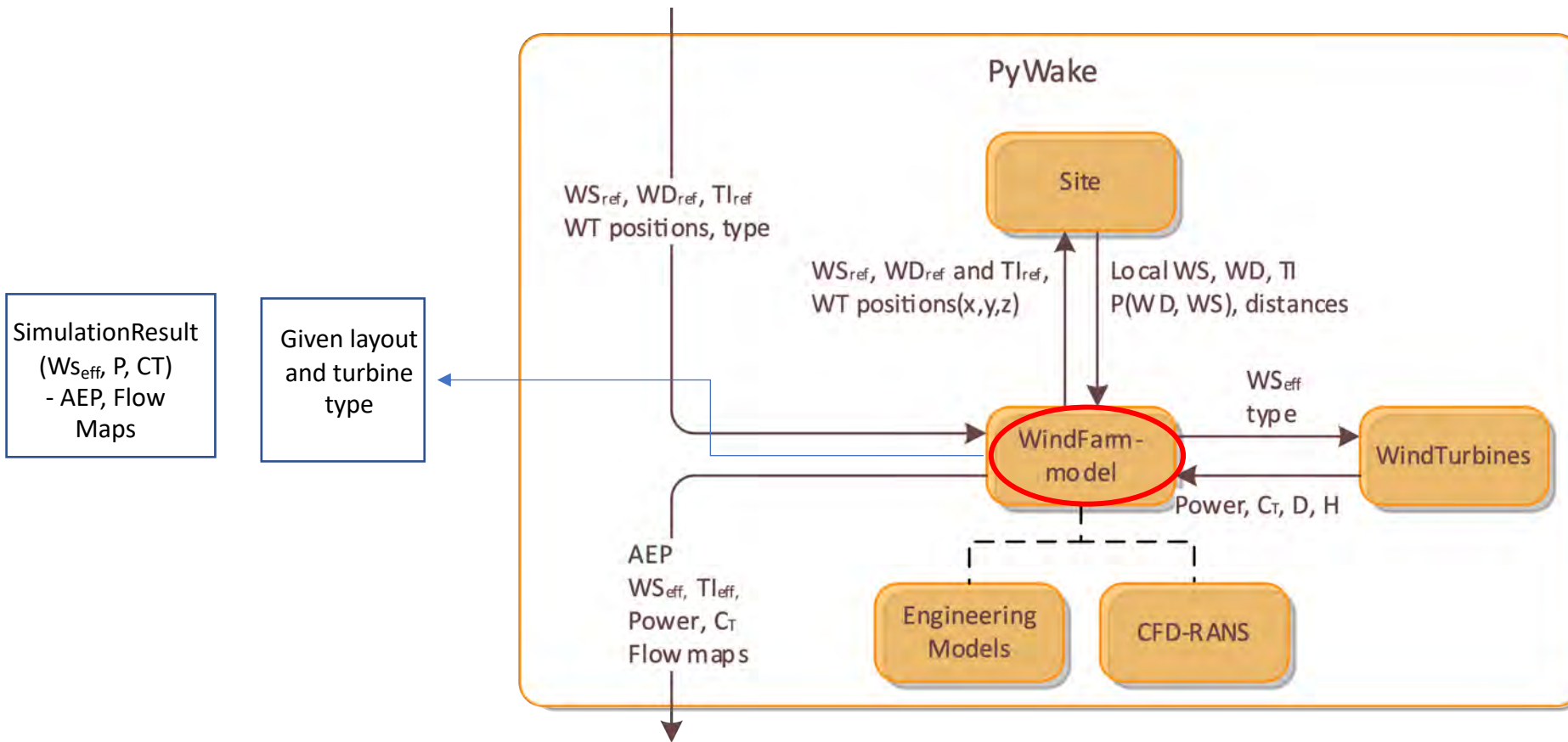


# PyWake: Wind Turbines

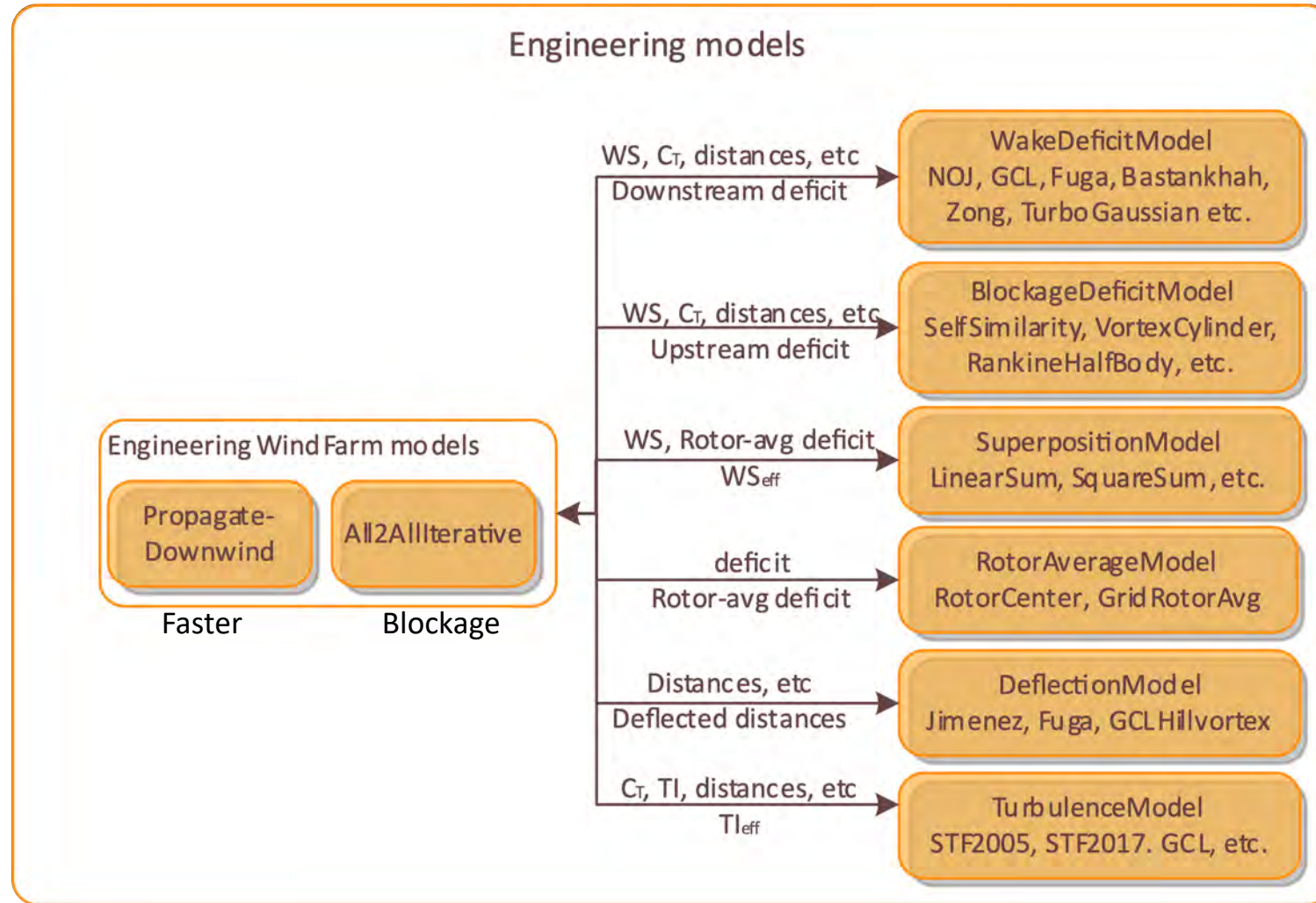
- Power and CT curves
- Multidimensional Power-CT dependence



# PyWake Overview

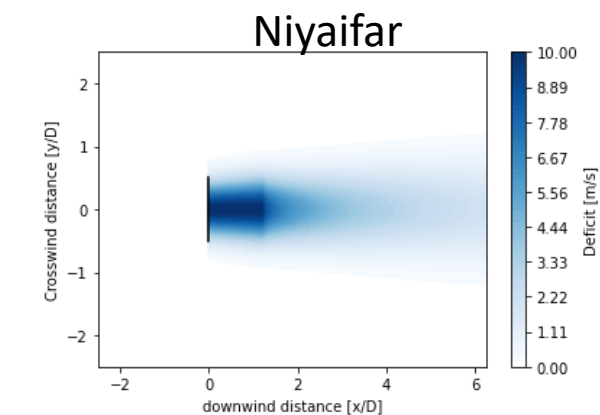
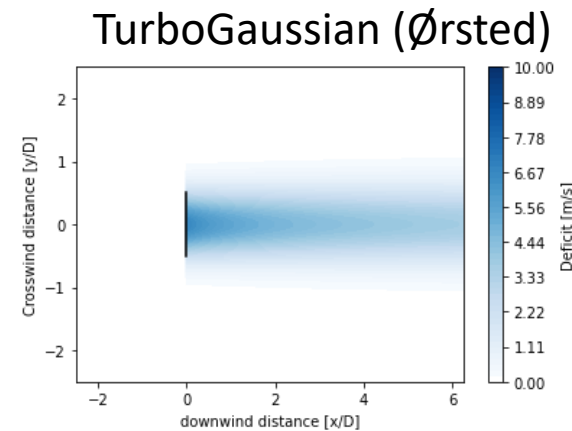
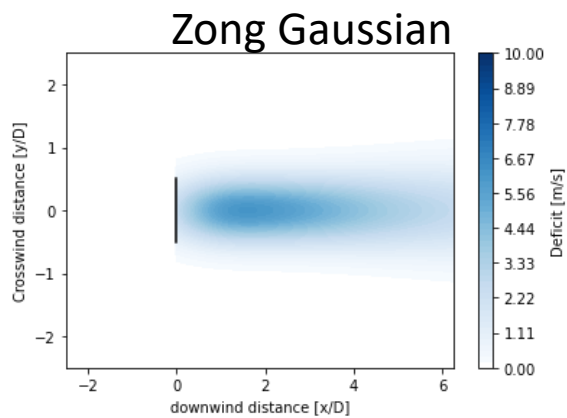
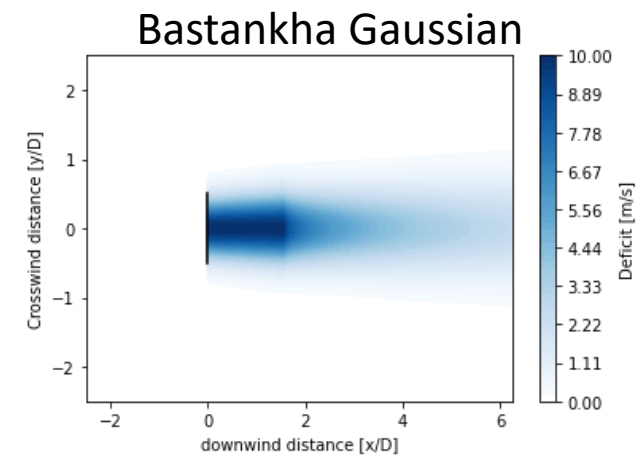
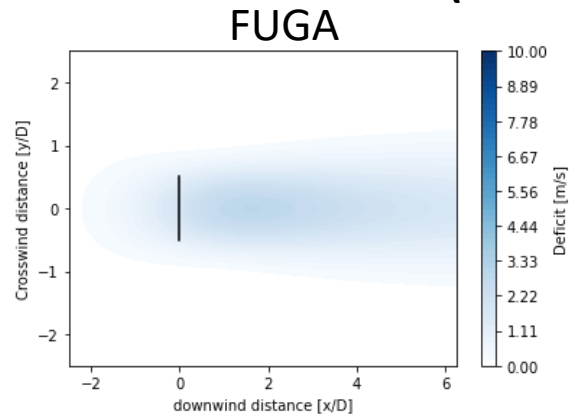
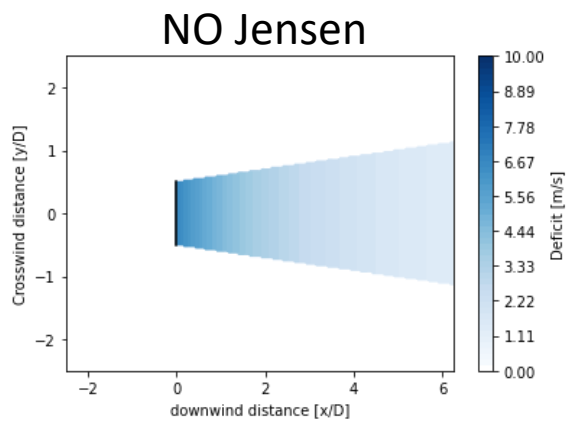


# PyWake: Wind Farm Model



# PyWake: Wake Models

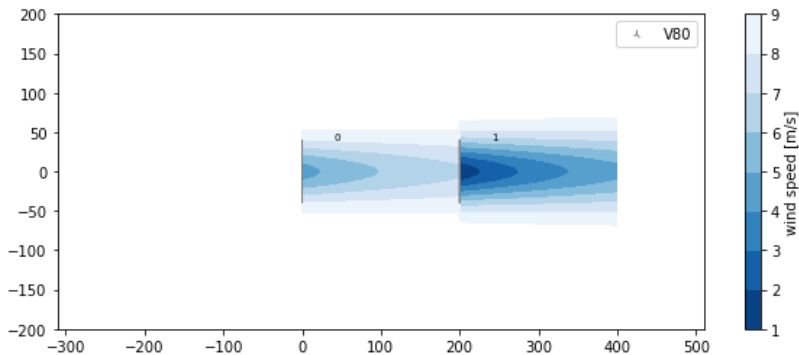
- Engineering Models, Linearized CFD (FUGA), and CFD Lookup tables



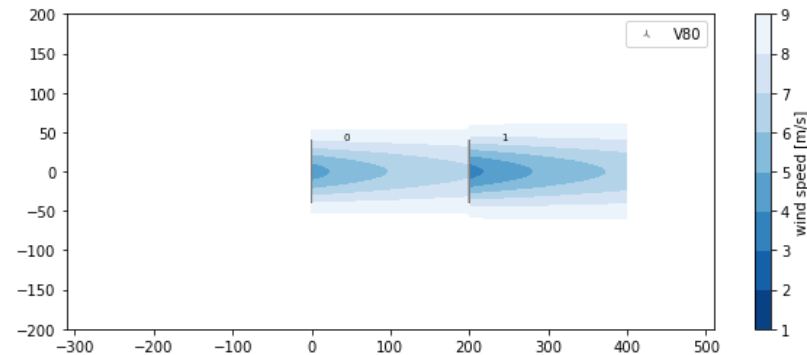
# PyWake: Wake Models

- Wake deficit: Superposition models

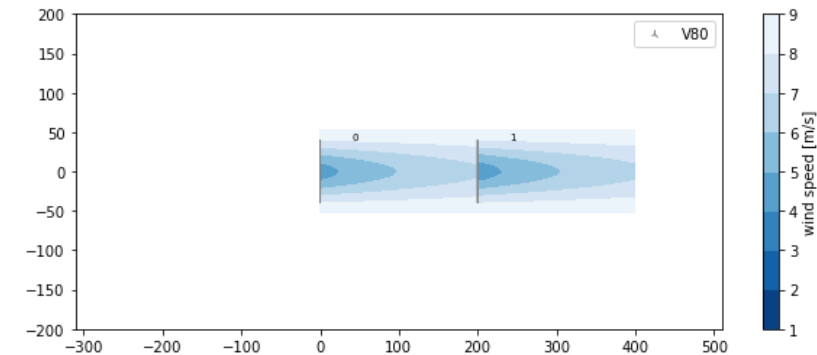
Linear superposition



Quadratic superposition

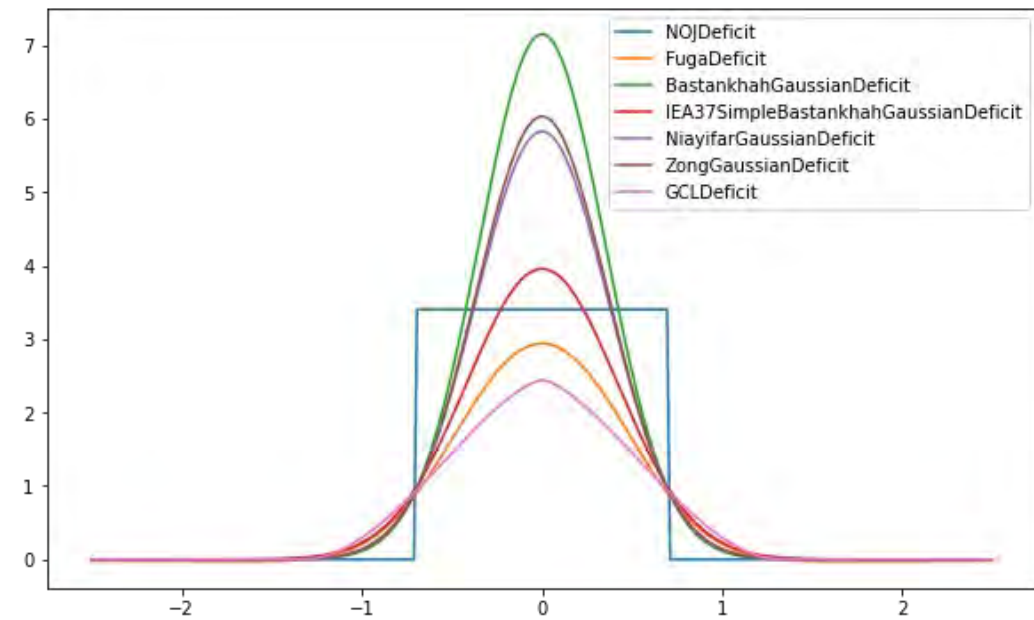
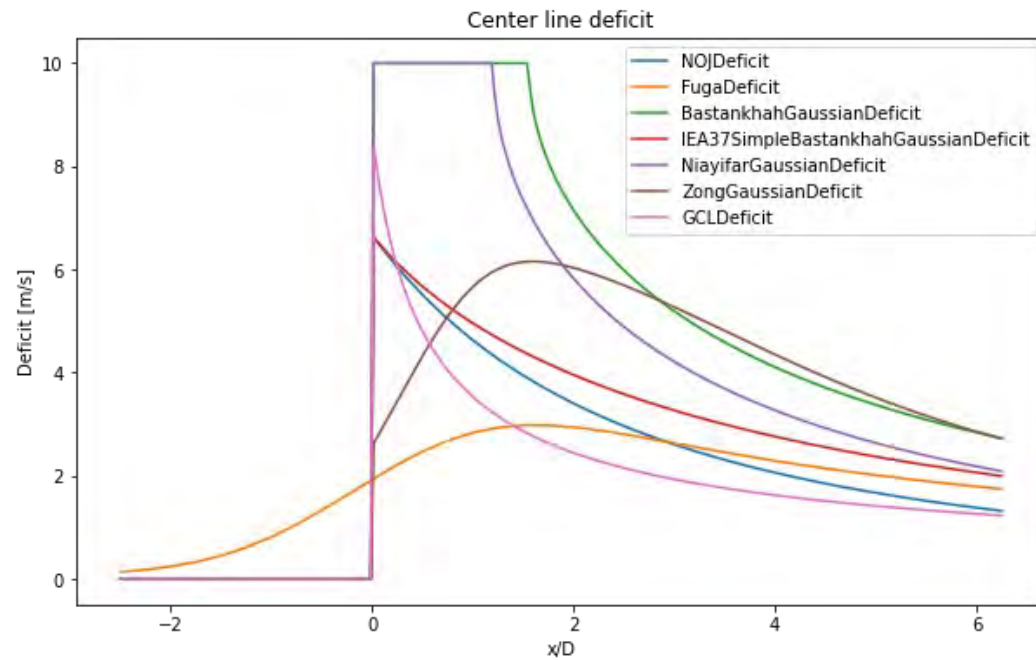


Maximum sum superposition



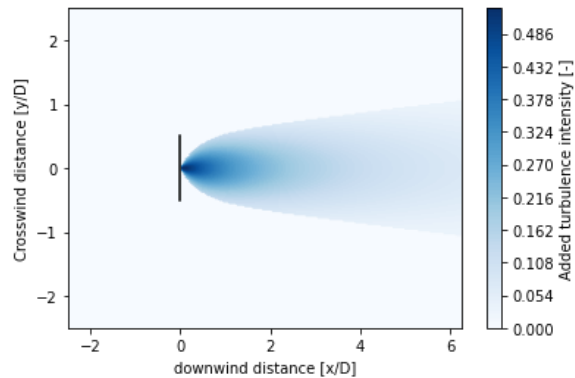
# PyWake: Wake Models

- Comparison on the wake deficit models

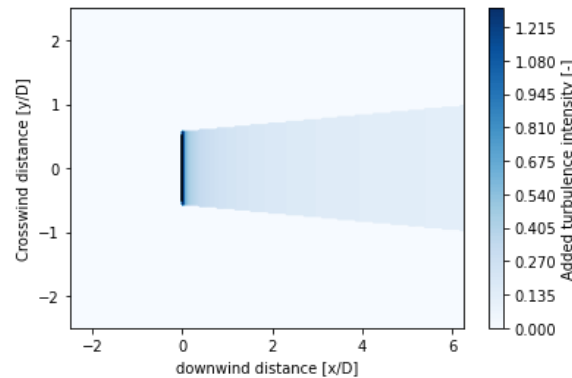


# PyWake: Wake Turbulence Models

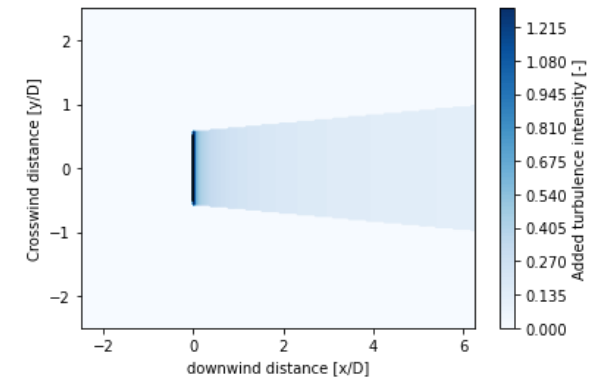
- Engineering Models



STF2005 / STF2017:  
Based on Frandsen (IEC61400-1)  
model



Gunner Larsen

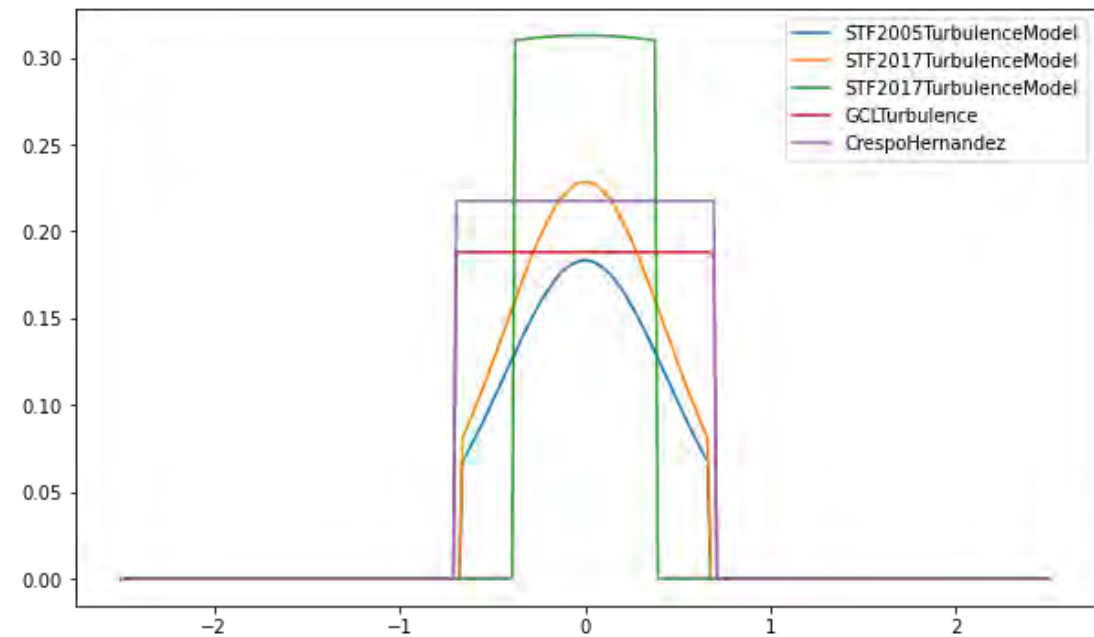
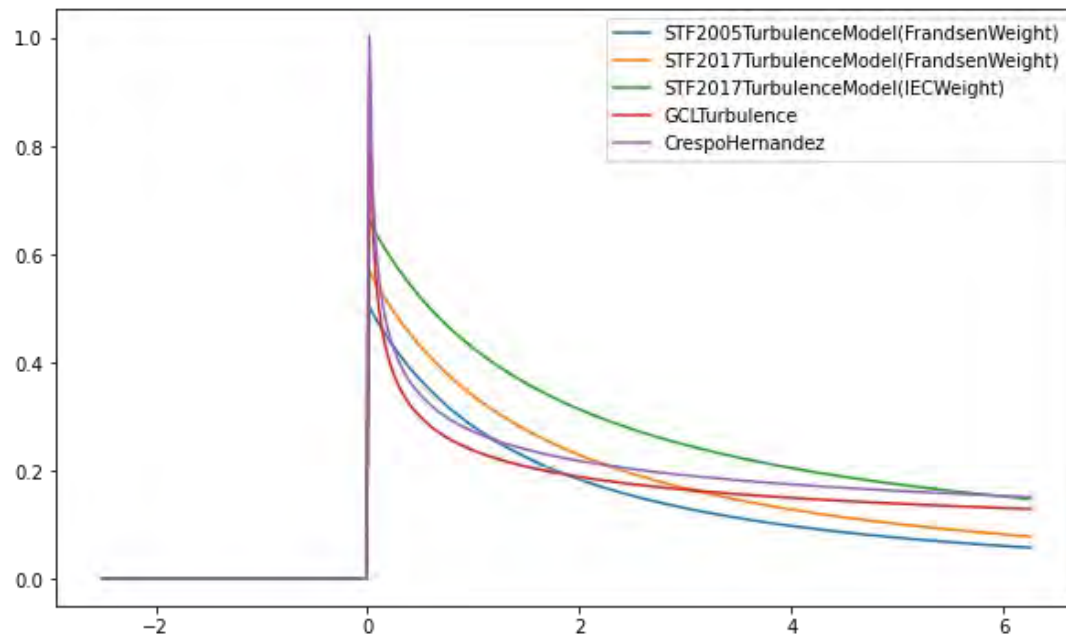


Crespo / Hernandez



# PyWake: Wake Turbulence Models

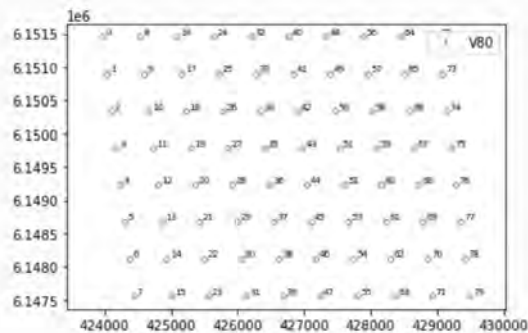
- Engineering Models



# PyWake: Wind Farm Modeling

- User provides site, wind turbine, layout and engineering models. PyWake returns AEP and flow field characteristics.

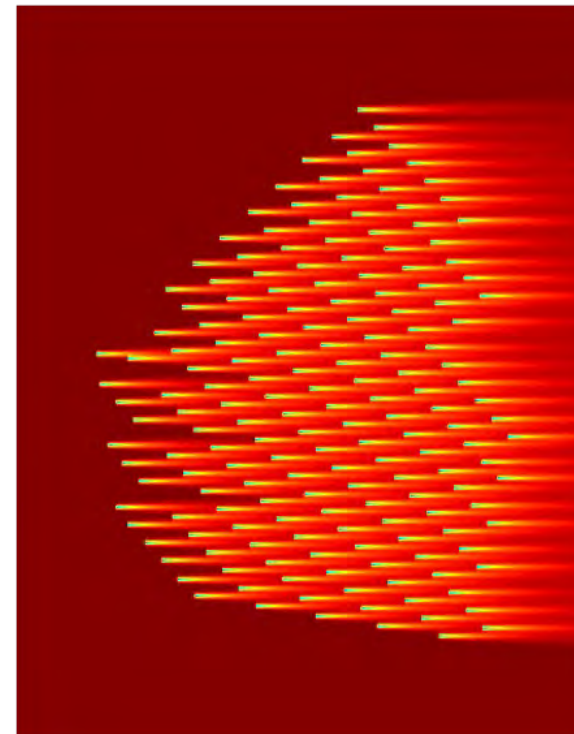
Horns Rev 1 Farm



Calculate AEP

```
[6]: # Original AEP
aep_ref = windFarmModel(wt_x,wt_y).aep().sum()
print (aep_ref)

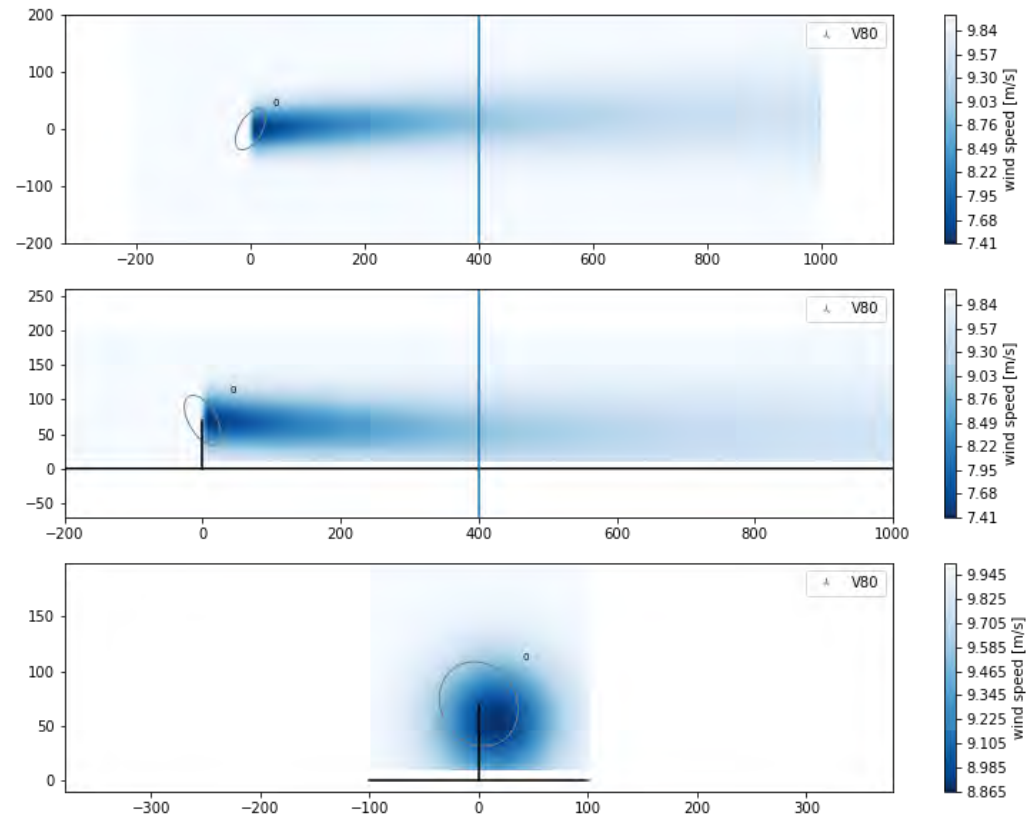
<xarray.DataArray 'AEP [GWh]' (>)
array(682.04072529)
```



**EAST ANGLIA**  
- Wake velocity  
flow field

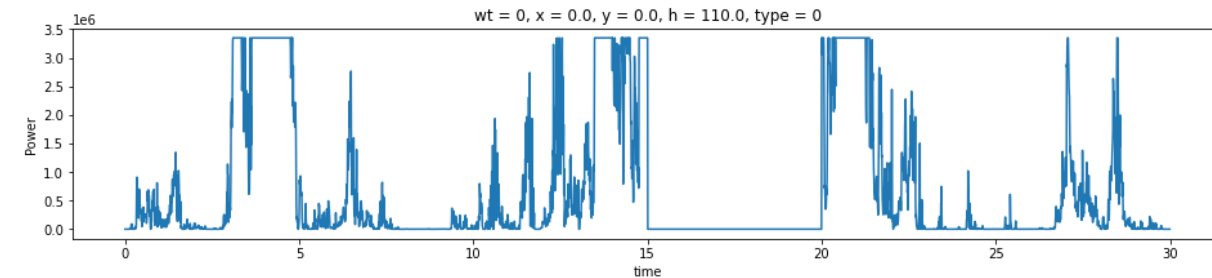
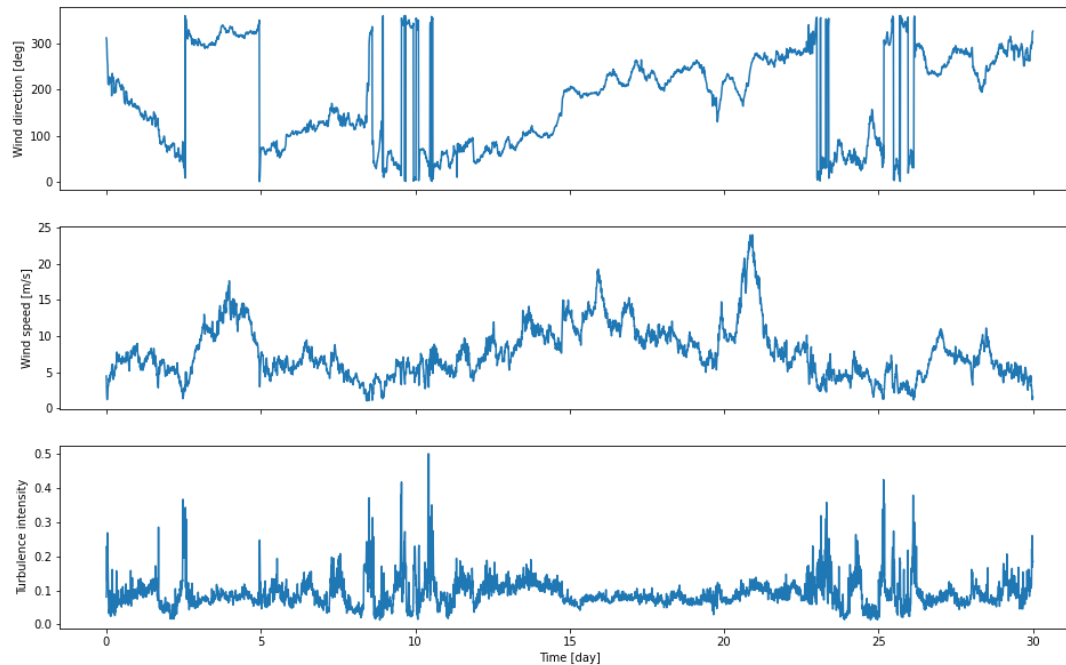
# PyWake: Additional Capabilities

- Yaw and Tilt angle



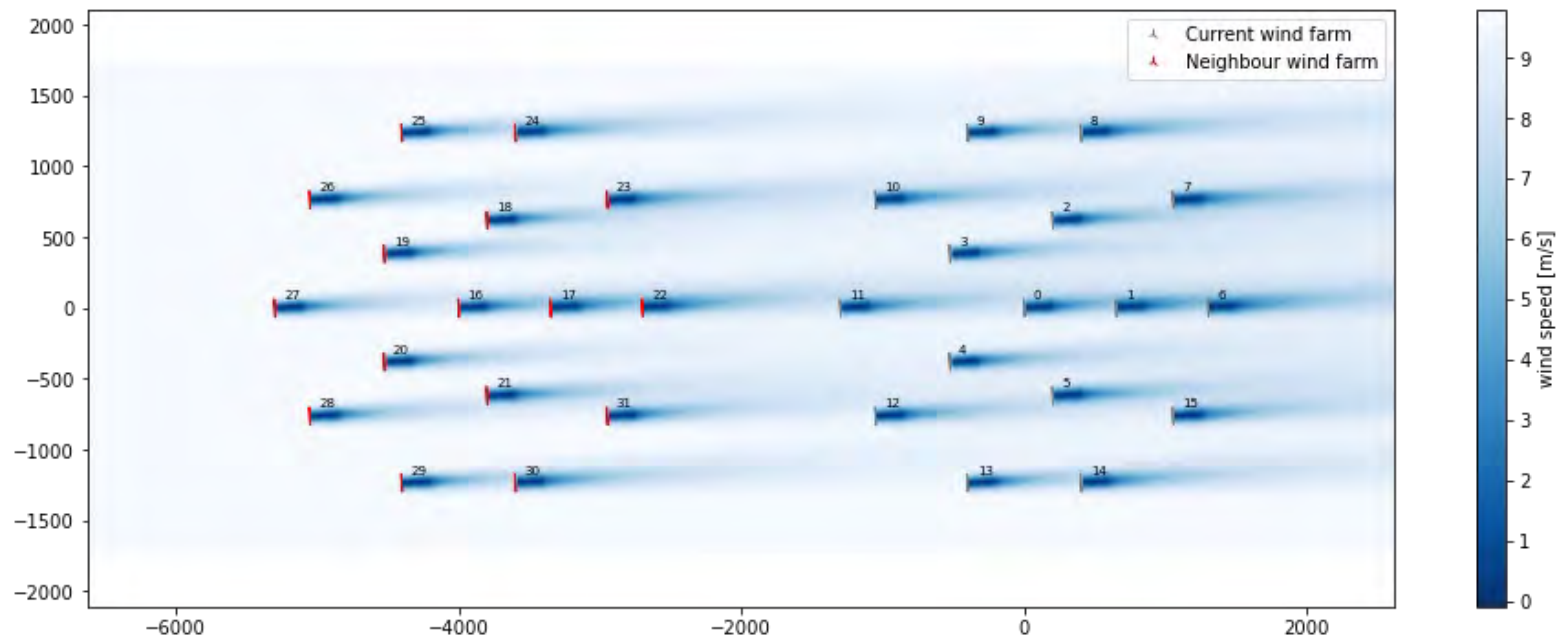
# PyWake: Additional Capabilities

- Time-series analysis (e.g., wind speed, wind direction, air density, TI) for AEP computations. Idling periods can be taken into consideration.

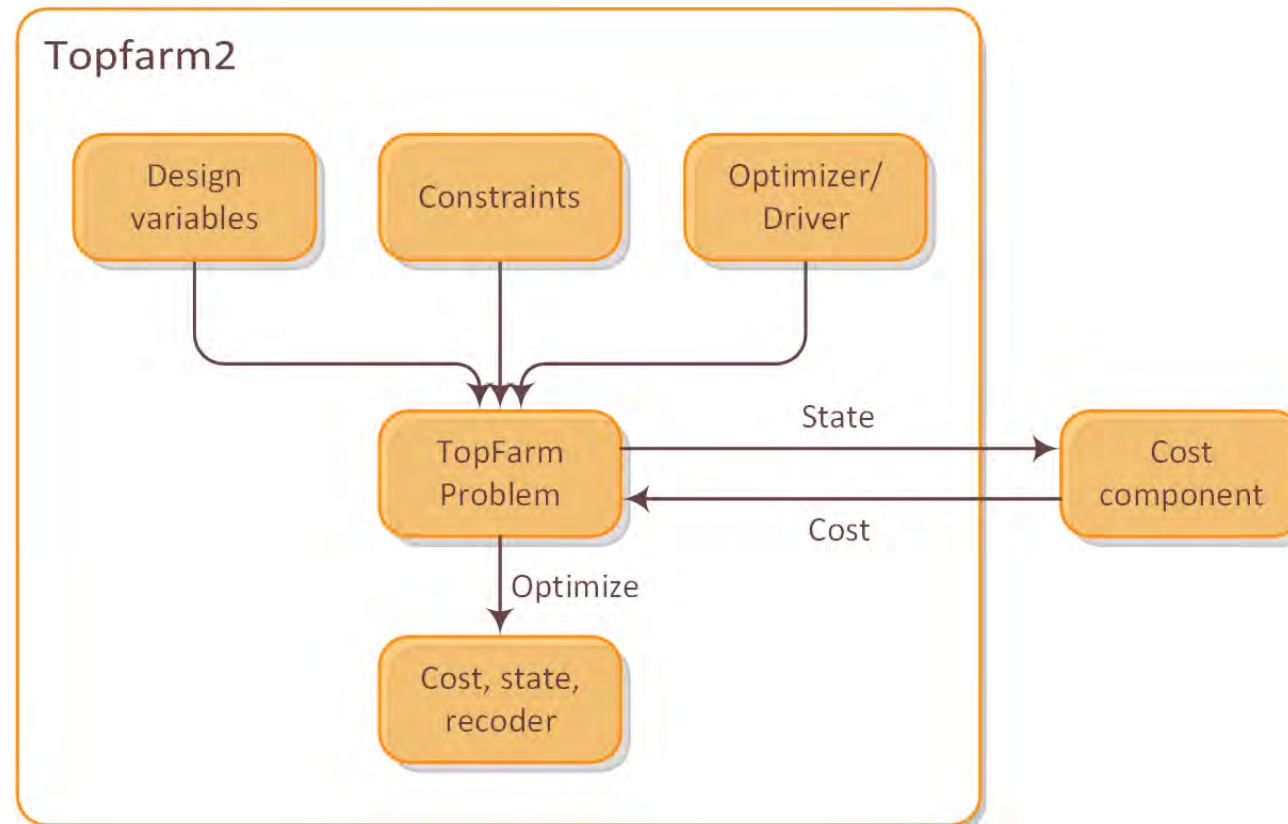


# PyWake: Additional Capabilities

- Farm to farm interaction. Possibility to couple with mesoscale models



# TOPFARM



# Questions?

- On behalf of DTU Wind (SYS Section), thank you for attending!