Spring 2015

Offshore Turbine Arrays: Numerical Modeling and Experimental Validation

Thomas Kroll
University of New Hampshire - Main Campus, tby5@unh.edu

William Hall
University of New Hampshire - Main Campus, whb7@unh.edu

Ian Gagnon
University of New Hampshire - Main Campus, ifh2@unh.edu

Offshore Turbine Arrays: Numerical Modeling and Experimental Validation

By Ian Gagnon, William Hall, Thomas Kroll

Faculty Advisor: Martin Wosnik
Graduate Student Advisors: John James Turner V, Ivaylo Nedyalkov, Pete Bachant

Sea Grant
New Hampshire

Table of Contents

Abstract

The interaction between wind turbines in a large wind farm needs to be better understood to reduce array losses and improve energy production. A numerical test bed for an array of offshore wind turbines was developed in the open-source computational fluid dynamics (CFD) framework *OpenFOAM*. It provides a computational tool which can be used in combination with physical model turbine array studies in the Flow Physics Facility (FPF) at UNH as well as other test facilities.

Turbines were modeled as actuator disks with turbulence sources to reduce computational cost. Both k-ε and k-ω SST turbulence models were utilized to capture the flow in the near-wall, wake, and free stream regions.

Experimental studies were performed in the FPF to validate the numerical results and to provide realistic initial and boundary conditions, for example turbulent boundary layer inlet velocity profiles. Mesh refinement and boundary condition studies were performed. Numerical simulations were executed on a custom-built server, designed to be the head node of a future CFD cluster. The entire project was built on open-source software to facilitate replication and expansion. The numerical model provides building blocks for simulations of large wind turbine arrays, computational resources permitting.

The numerical model currently replicates a three by one array of wind turbines in the FPF, and provides insight into the array fluid dynamics with limitations.

Acronyms

Abbreviation	*Description*
BC	**Boundary Condition**
BEM	**Blade Element Momentum**
BL	**Boundary Layer**
CFD	**Computational Fluid Dynamics**
ECC	**Error-correcting code memory**
D	**Normalized Diameter (0.25m for the model wind turbine)**
FPF	**Flow Physics Facility**
FS	**Freestream**
OS	**Operating System**
PCI	**Peripheral Component Interconnect**
RAM	**Random-access memory**
RANS	**Reynolds-Averaged Navier-Stokes equations**
SIMPLE	**Semi-Implicit Method for Pressure Linked Equations**

Variables

Variable	*Description*	*Units*
C	**Turbulence injection constant**	$-$
C_p	**Coefficient of power**	$-$
C_t	**Coefficient of thrust**	$-$
ε	**Dissipation of kinetic energy**	$\frac{m^2}{s^3}$
I	**Turbulence intensity**	$-$
κ	**Von Kármán constant**	$-$
k	**Turbulence kinetic energy**	$\frac{m^2}{s^2}$
l or ℓ	**Length scale**	m
μ	**Dynamic viscosity**	$\frac{Ns}{m^2}$
μ_t	**Eddy viscosity**	$\frac{Ns}{m^2}$
ν	**Kinematic viscosity**	$\frac{m^2}{s}$
Ω	**Rate of dissipation of kinetic energy**	$\frac{1}{s}$
ρ	**Density**	$\frac{kg}{m^3}$
$P\ or\ p$	**Pressure**	$\frac{N}{m^2}$
p'	**Fluctuating pressure component**	$\frac{N}{m^2}$
$\bar{p}$	**Averaged pressure component**	$\frac{N}{m^2}$
Re	**Reynolds number**	$-$
$\overline{S_{ij}}$	**Reynolds-averaged strain rate tensor**	$\frac{kg}{m \cdot s^2}$
τ_w	**Shear stress at the wall**	$\frac{N}{m^2}$

τ_{ij}^{R}	**Reynold Stress tensor**	$\frac{kg}{m \cdot s^2}$
ϑ	**Velocity scale**	$\frac{m}{s}$
t	**Time**	s
T	**Time interval**	s
U^{+}	**U plus**	$-$
U_o	**Incoming velocity to actuator disk**	$\frac{m}{s}$
u_T	**Shear velocity**	$\frac{m}{s}$
v_i	**Velocity vector in Cartesian index notation**	$\frac{m}{s}$
v_i'	**Fluctuating velocity component**	$\frac{m}{s}$
$\overline{v_\imath}$	**Averaged velocity component**	$\frac{m}{s}$
$\vec{V}$	**Velocity Vector**	$\frac{m}{s}$
Y	**Height**	m
Y^{+}	**Y plus**	$-$

Motivation

Since the 1990’s, wind turbines have begun to be installed in large wind farms or arrays. These arrays are advantageous due to the fact that they cluster many turbines into an area of high wind resource and aim to use less space. In addition, the transmission of power generated is simpler from an array compared to a distributed network of individual turbines. However, there are some unwanted attributes to arranging wind turbines in arrays. An example of this is exhibited on *Figure* (1) which shows the coefficient of power or C_p for model wind turbines in an array.

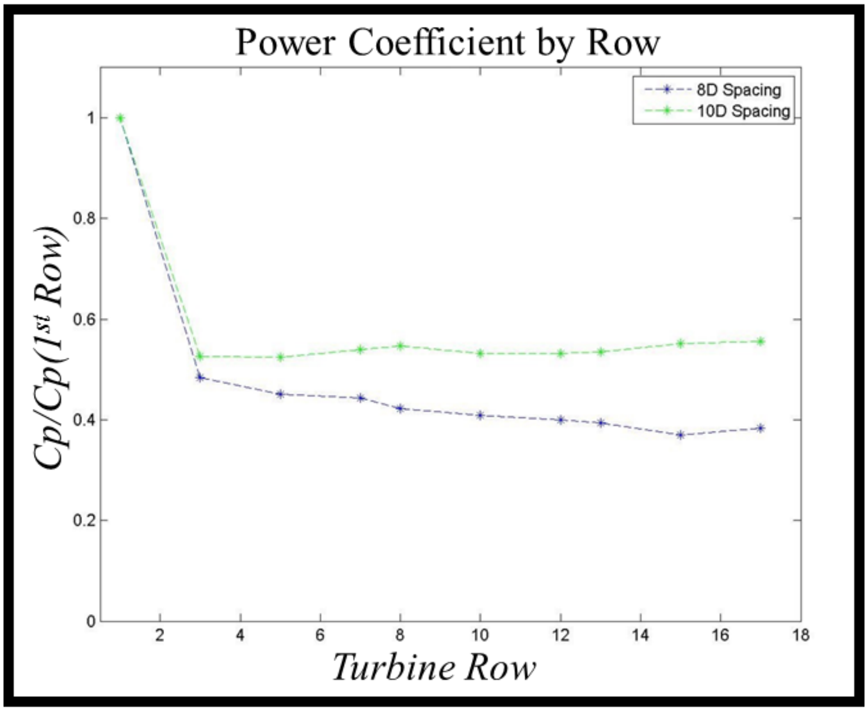


Figure 1: Plot of normalized coefficient of power by row of model wind turbine array. The Cp values are normalized with the Cp of the first model wind turbine. You can observe that the normalized Cp drops significantly after the first wind turbine, meaning that there is a drop in overall array power production. [5]

All wind turbines create a turbulent wake downstream of their rotor, in addition to a decrease in the velocity and pressure of the flow. In arrays, downstream turbines will operate in the wake of upstream turbines. Due to the decreased velocity in these wakes, it has been shown that the second downstream turbine in an array’s performance can be decreased by at least 20% compared to the first turbine based on wind direction [6].

There are large financial losses that stem from these array wake effects. Take for example the largest wind farm in the US, the Alta Wind Energy Center in Kern County California. This array has an installed rated capacity of 1.32 GW. However, there is a large drop in the capacity factor because downstream turbine performance can be decreased by at least 20% [1]. This is one of the major contributing factors to why onshore wind farms typically have a capacity factor of around 0.2-0.3.

The United States Department of Energy (DOE) has set the goal of installing enough wind energy capacity such that 20% of the nation's electrical energy comes from wind energy by 2030. This would represent an approximate installed rated capacity of 305 GW. Every 1% array loss eliminated from this future 305 GW represents approximately 1 trillion USD in electricity generated.

Decreasing the loss of energy generation is not the only financial factor that will benefit from studying array effects. The increased turbulence in turbine wakes causes cyclic loading on downstream turbine rotors. These cyclic loads promote gearbox wear and blade cracking. These effects reduce turbine life and increase the cost of operating and maintaining a wind farm.

Background

Since early 2011, studies have been performed on wind turbines at UNH's Flow Physics Facility (FPF). The FPF is the world's largest boundary layer (BL) wind tunnel. The test section of the tunnel is 6 m by 2.7 m by 72 m. It can reach velocities of up to 14 m/s. Its typical freestream turbulence intensity is between 0.2 and 0.5%. Two 400 HP fans drive the flow and enable the tunnel to naturally grow a turbulent boundary layer approximately 1 m thick at the end of the test section. These characteristics, especially the tunnels long test section allow for very realistic studies of wind turbine arrays to be performed in it.

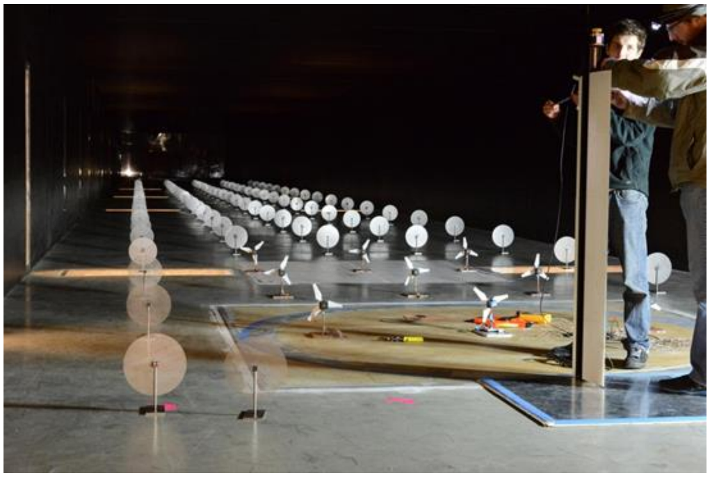


Figure 1: Experiment in FPF with model wind turbines and porous disks [5]

Those studying fluid dynamics approach increasing their knowledge through theoretical understanding of their problem, experimentation and numerical modeling. Experiments are crucial in simulating real world phenomena that aren't easily numerically recreated. Oftentimes, experiments can yield more convincing data than numerical results. Numerical models, on the other hand, are oftentimes cheaper and less time consuming to modify. In addition, unlike experiments, optimization techniques can be more easily used in conjunction with numerical models. Good modern researchers take advantage of both experimental and numerical tools to achieve the truest results with the least expense.

Currently there is no numerical model being utilized in conjunction with experiments in the FPF. Our team set out to develop a numerical model of an array of wind turbines in the FPF. Reasons for this are to increase the efficiency of the current experimental work being performed in the wind tunnel as well as to provide a tool that could be used to predict and validate experimental results here in the FPF as well as other facilities.

Theory

Wind Turbines

Typical horizontal axis wind turbines (HAWT) use three blades with an axis of rotation in the same direction as the wind. The wind turbine is composed of a foundation, tower, blades, and a nacelle where the generator and sometimes a gearbox is housed. The wind creates forces on the blades which turn the turbine rotor which then spin the generator to produce electricity. Offshore wind generation has increased in popularity due to the large and consistent wind resource off many coasts. The global offshore wind capacity is approximately 7-8 GW. Despite the potential value of this resource the US has yet to install offshore wind turbine arrays. A map of the wind resources and transmission lines in the United States can be seen below in *Figure*

(3).

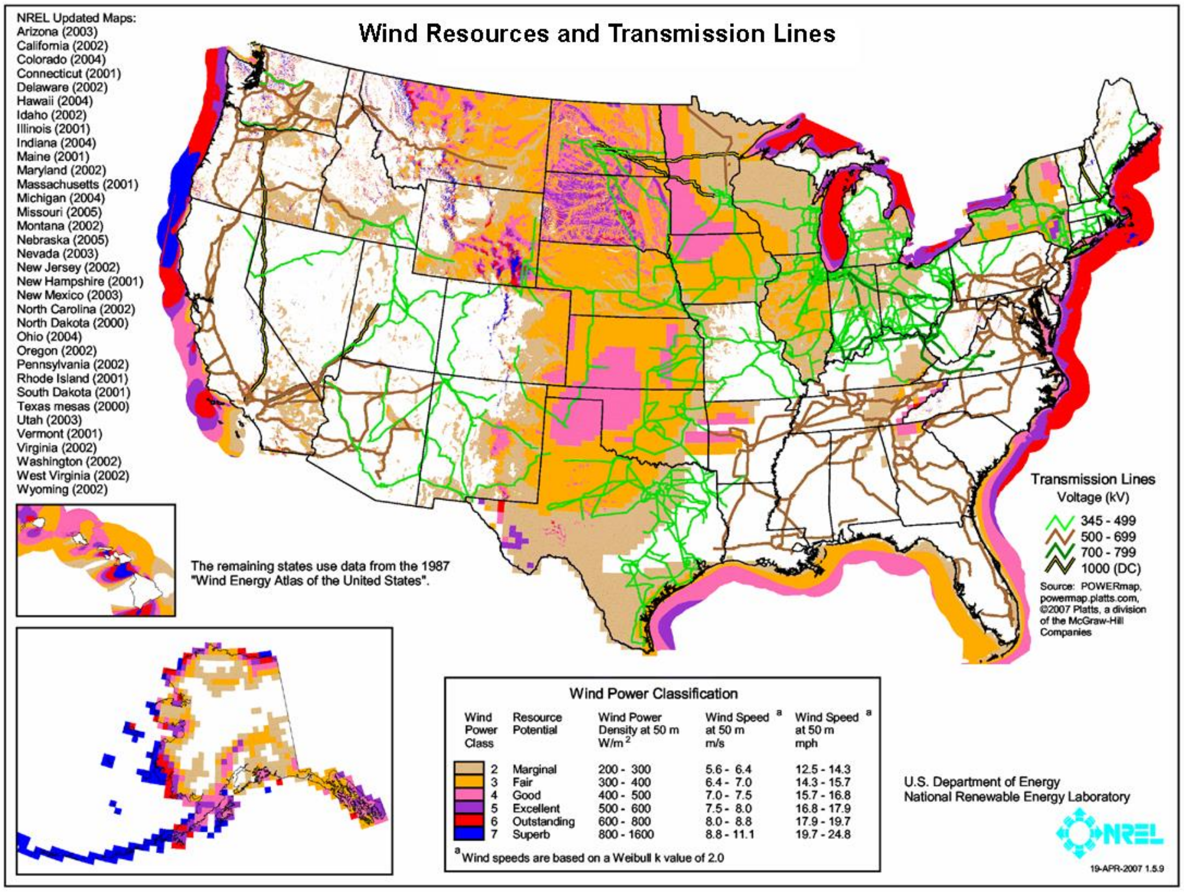

Wind Resources and Transmission Lines
NREL Updated Maps:
Arizona (2003)
California (2002)
Colorado (2004)
Connecticut (2001)
Delaware (2002)
Hawaii (2004)
Idaho (2002)
Illinois (2001)
Indiana (2004)
Maine (2001)
Maryland (2002)
Massachusetts (2001)
Michigan (2004)
Missouri (2005)
Montana (2002)
Nebraska (2005)
Nevada (2003)
New Jersey (2002)
New Hampshire (2001)
New Mexico (2003)
North Carolina (2002)
North Dakota (2000)
Ohio (2004)
Oregon (2002)
Pennsylvania (2002)
Rhode Island (2001)
South Dakota (2001)
Texas mesas (2000)
Utah (2003)
Vermont (2001)
Virginia (2002)
Washington (2002)
West Virginia (2002)
Wyoming (2002)
The remaining states use data from the 1987 "Wind Energy Atlas of the United States".
Transmission Lines
Voltage (kV)
345 - 499
500 - 699
700 - 799
1000 (DC)
Source: POWERmap, powermap.platts.com, ©2007 Platts, a division of the McGraw-Hill Companies
Wind Power Classification
Wind Power Class
Resource Potential
Wind Power Density at 50 m W/m²
Wind Speed at 50 m m/s [a]
Wind Speed at 50 m mph [a]
2 Marginal 200 - 300 5.6 - 6.4 12.5 - 14.3
3 Fair 300 - 400 6.4 - 7.0 14.3 - 15.7
4 Good 400 - 500 7.0 - 7.5 15.7 - 16.8
5 Excellent 500 - 600 7.5 - 8.0 16.8 - 17.9
6 Outstanding 600 - 800 8.0 - 8.8 17.9 - 19.7
7 Superb 800 - 1600 8.8 - 11.1 19.7 - 24.8
[a] Wind speeds are based on a Weibull k value of 2.0
U.S. Department of Energy
National Renewable Energy Laboratory
NREL
19-APR-2007 1.5.9

Figure 2: Wind Resources map with transmission lines [8]

Wind turbines convert the motion of the wind into rotational shaft work and in turn into electrical energy. In doing so the wind turbine removes momentum from the flow. The wind turbine blades' interaction with the flow also introduces turbulence into its wake.

Boundary Layer Theory

Arrays of wind turbines operate within the atmospheric boundary layer. A boundary layer is a layer of flow nearest to a bounding wall where the time averaged velocity increases from zero at the wall on bounding surface to its free stream value moving away from the wall. The boundary layer thickness is usually defined as where the velocity reaches 0.99 of the free stream fluid's velocity. At either high enough velocities or long enough lengths, the top, faster layers of the boundary layer begin to tumble down onto the bottom, slower layers. This creates a swirling effect which is known as turbulence.

The air flow on the earth forms what is called the Atmospheric Boundary Layer (ABL). This layer of air is more than a kilometer in height. Inside the atmospheric boundary layer and any boundary layer, the Navier-Stokes equations can be solved so certain dependencies can be determined. To make predictions about these fluid layers, non-dimensionalized terms are used to describe the velocity of the fluid and the distance from the bounding wall. In doing this the Law of the Wall is created.

Y+ and U+

The Law of the Wall relates the two dimensionless parameters $Y^+$and U^+ within the boundary layer of a horizontal fluid flow. $Y^+$is defined as the dimensionless distance from the wall by multiplying the actual coordinate distance y by the shear velocity u_T at the wall and dividing by the kinematic viscosity ν of the fluid. U^+ is the dimensionless flow velocity. This dimensionless flow velocity is defined differently depending on the distance from the wall. The equation for Y^+can be seen below in *Equation* (1) [7]

$$Y^+ = \frac{yu_T}{\nu} \tag{1}$$

where y is the height away from the wall, $u_T = \sqrt{\frac{\tau_w}{\rho}}$is the shear velocity with τ_w being the shear stress at the wall and ρ the fluid density , ν is the kinematic viscosity of the fluid, and u is the incoming horizontal flow velocity. Within the closest region of the flow to the wall, known as the Viscous Layer, where $Y^+ < 5$, U^+ is considered equal to Y^+. The region of the flow above where Y^+ equals 30, is called the Log-Law Region Inner Layer. Within the layer U^+ is defined by *Equation* (2)

$$U^+_{Y^+<5} = Y^+ \tag{2}$$

$$U^+_{Y^+>30} = \frac{1}{\kappa}\ln(Y^+) + C^+$$

where κ is the Von Kármán constant, and C^+ is a constant. Between the Y^+values of 5 and 30, there is the Buffer Layer. Within this region U^+ is not defined by either of the two ways in the Viscous Layer or the Log-Law Region Inner Layer. Therefore it must be approximated using the values where Y^+ equals 5 and Y^+ is above 30 as shown in *Equations* (1) and (2).

Computational Fluids Dynamics (CFD)

Navier-Stokes Equations

The Navier-Stokes equations describe the motion of a fluid particle. They are derived from Newton's second law. For our case, we utilize the equations under the assumptions that the fluid being described is incompressible and Newtonian. The Navier-Stokes conservation of mass and momentum equations can then be represented by *Equation* (3) and *Equation* (4) respectively [3]

$$\nabla \cdot \vec{V} = 0 \tag{3}$$

$$\rho\left(\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla\vec{V}\right) = -\nabla P + \mu\nabla^2\vec{V} \tag{4}$$

where ρ is the density of the fluid, $\vec{V}$ is the velocity field of the fluid, P is the pressure, t is time, and μ is the dynamic viscosity of the fluid.

Turbulence

Past the critical Reynolds number, there is a radical change in the flow characteristics where by the flow acts in a random and chaotic manner. The flow becomes unsteady and the flow properties vary randomly. This is what is known as a turbulent flow and is one of the characteristics of the flow for the numerical model that was built [7].

Reynold-Averaged Navier Stokes Equations (RANS)

From the Navier-Stokes momentum equation shown above in *Equation* (4), averaging techniques can be used to simplify turbulent flow. In doing this, the velocity component of the turbulent flow is broken down into averaged $\overline{v_i}$ and fluctuating v_i' terms. The velocity and pressure are represented respectively as

$$v_i = \overline{v_i} + v_i'$$

$$p = \bar{p} + p'$$

where the mean values are denoted with an over bar and the fluctuating terms with a prime. With time averaging, the mean velocity term is obtained using *Equation* (5) below. Note that for our case, we use the time average because our problem is steady state. The Navier-Stokes equations are ensemble averaged for unsteady problems [3]

$$\overline{v_i} = \lim_{T\to\infty} \frac{1}{T}\int_t^{t+T} v_i dt \tag{5}$$

where T is the time interval and t is the time. Since this mean velocity is integrated with time, it does not vary in time, but only in space. From this time average, the Reynolds-Averaged Navier-Stokes equations (RANS) are obtained. The averaged equations for mass and momentum conservation are shown below in *Equations* (6) and (7) below in Cartesian index notation.

$$\frac{\partial \overline{v_i}}{\partial x_i} = 0 \tag{6}$$

$$\rho\left(\frac{\partial \overline{v_i}}{\partial t} + \overline{v_j}\frac{\partial \overline{v_i}}{\partial x_j}\right) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\overline{\tau_{ij}} - \tau_{ij}^R\right) \tag{7}$$

These equations are the same as the original Navier-Stokes equations besides the additional term known as the Reynolds-stress tensor shown in *Equation* (8).

$$\tau_{ij}^R = -\rho\overline{v_i' v_j'} = -\rho\left(\overline{v_i v_j} - \overline{v_i}\,\overline{v_j}\right) \tag{8}$$

This term in the RANS momentum equation represents the transfer of fluid momentum caused by the turbulent fluctuations. The laminar viscous stresses in the fluid are solved for using the Reynolds-averaged velocity components represented by *Equation* (9).

(9)

$$\overline{\tau_{ij}} = 2\mu\overline{S_{ij}} = \mu(\frac{\partial\overline{v_i}}{\partial x_j} + \frac{\partial\overline{v_j}}{\partial x_i})$$

Reynold Stresses

The Reynolds-stress is an extra term in the Navier-Stokes momentum equation after being averaged. The Reynold-stress tensor can be seen in *Equation* (10). In three dimensions there consist nine components and six independent, unknown terms. The subscripts 1, 2, and 3 represent the x, y, and z components respectively [3]:

$$\rho\overline{v_i'v_j'} = \begin{bmatrix} \rho\overline{(v_1')^2} & \rho\overline{v_1'v_2'} & \rho\overline{v_1'v_3'} \\ \rho\overline{v_2'v_1'} & \rho\overline{(v_2')^2} & \rho\overline{v_2'v_3'} \\ \rho\overline{v_3'v_1'} & \rho\overline{v_3'v_2'} & \rho\overline{(v_3')^2} \end{bmatrix} \qquad (10)$$

RANS Turbulence Modeling

For turbulence modeling through the RANS equations, the focus is on the effects of turbulence on the mean flow. Therefore, to be able to compute turbulent flows, it is important that the Reynold-stresses are modelled to close out the system of equations. RANS turbulence models are named after the number of additional transport equations that are needed to be solved together with the RANS equations. We chose to work with what are known as Two Equation turbulence models.

Eddy Viscosity Model

RANS turbulence models are based on the observations made by Boussinesq in 1877 known as the Boussinesq approximation. This idea states that the Reynold-stresses can be broken down into shear and normal turbulent stress. The turbulent shear stress is related to the mean rate of strain or fluid deformation rate $\overline{S_{ij}}$ like in laminar flow similar to *Equation* (9) except for the use of what is known as eddy viscosity μ_t. Eddy viscosity μ_t is a function of the flow conditions. The turbulent normal stress is related to the turbulent kinetic energy k of the flow [3]. The turbulent stresses can then be represented together in *Equation* (11):

$$\tau_{ij}^R = -\rho\overline{v_i'v_j'} = 2\mu_t\overline{S_{ij}} - \frac{2}{3}\rho k\delta_{ij} \qquad (11)$$

δ_{ij} is the Kronecker delta where $\delta_{ij} = 1\ if\ i = j\ and\ \delta_{ij} = 0\ if\ i \neq j$

The turbulent kinetic energy is $k = \frac{1}{2}\overline{v_i'v_i'} = \frac{1}{2}(\overline{v_1'^2} + \overline{v_2'^2} + \overline{v_3'^2})$

The k-ε model

The k-ε model uses two modeling equations to describe the flow and is thus known as a Two Equation turbulence model. These equations represent the turbulence kinetic energy k of the flow and its dissipation rate ε. These parameters are used to define the velocity scale ϑ and

(12)

the length scale l of the large-scale turbulence. The equations describing these parameters can be seen below in *Equations* (12) and (13) [7].

$$\vartheta = k^{\frac{1}{2}}$$

$$l = \frac{k^{\frac{3}{2}}}{\varepsilon} \tag{13}$$

Even though ε is defined as the small eddy kinetic energy dissipation rate, it can still be used to characterize the large eddy scale l since at high Reynolds numbers the rate at which energy is extracted from the flow matches that of smaller eddies. By using dimensional analysis the eddy viscosity can be solved for using

$$\mu_t = C\rho\vartheta l = \rho C_\mu \frac{k^2}{\varepsilon}$$

where the variable C_μ is a predefined dimensionless constant. The final transport equations for the standard k-ε model are shown in *Equations* (14) and (15) below.

$$\frac{\partial(\rho k)}{\partial t} + div(\rho kU) = div\left[\frac{\mu_t}{\sigma_k} grad\ k\right] + 2\mu_t S_{ij} * S_{ij} - \rho\varepsilon \tag{14}$$

$$\frac{\partial(\rho\varepsilon)}{\partial t} + div(\rho\varepsilon U) = div\left[\frac{\mu_t}{\sigma_\varepsilon} grad\ \varepsilon\right] + C_{1\varepsilon}\frac{\varepsilon}{k} 2\mu_t S_{ij} * S_{ij} - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{15}$$

These equations for the kinetic energy and the dissipation rate of kinetic energy can be seen described in words below in *Equation* (16).

$$\begin{aligned} &Rate\ of\ change + Transport\ by\ convection \\ &\quad = Transport\ by\ diffusion + Rate\ of\ production - Rate\ of\ destruction \end{aligned} \tag{16}$$

By fitting data to a variety of turbulent flows, the constants C_μ, σ_k, σ_ε, $C_{1\varepsilon}$, and $C_{2\varepsilon}$ in the transport equations were given adjustable values. The values for these constants can be seen in *Table* (1).

Variable	C_μ	σ_k	σ_ε	$C_{1\varepsilon}$	$C_{2\varepsilon}$
Value	0.09	1.00	1.30	1.44	1.92

Table 1: Table of the adjustable constants for the standard k- ε model transport equations

The k-ε model tends to have high accuracies in free stream conditions. Near the wall, wall-functions are used to correlate the mean velocity, kinetic energy, and rate of dissipation of the flow to the shear stress at the wall at high Reynolds numbers. At lower Reynolds numbers modifications are made to include the molecular viscosity in the diffusion terms of the transport equations. Some of the adjustable constants are then multiplied by wall-damping functions. Even with these adjustments the k-ε model has problems near the wall due to the ε boundary condition. While getting closer to the wall, the rate of dissipation of kinetic energy rapidly increases and then settles on an unknown constant value.

The k-ω model

Unlike the k-ε in which the dissipation rate of kinetic energy, ε is used as the second transport equation. For the k-ω model, the specific dissipation or turbulence frequency $\omega = \frac{\varepsilon}{k}$ is used. The length scale and eddy viscosity are then defined as *Equations* (17) and (18) respectively.

$$l = \frac{k^{\frac{1}{2}}}{\omega} \qquad (17)$$

$$\mu_t = \frac{\rho k}{\omega} \qquad (18)$$

The Reynolds-stresses are then found using the Boussinesq expression, shown in *Equation* (11). For high Reynolds numbers the transport equations for k-ω model are described as *Equations* (19) and (20).

$$\frac{\partial(\rho k)}{\partial t} + div(\rho k U) = div\left[(\mu + \frac{\mu_t}{\sigma_k})grad\ (k)\right] + \left(2\mu_t S_{ij} * S_{ij} - \frac{2}{3}\rho k \frac{\partial U_i}{\partial x_j}\delta_{ij}\right) - \beta^{*}\rho \mathrm{k}\omega \qquad (19)$$

$$\frac{\partial(\rho \omega)}{\partial t} + div(\rho \omega U) = div\left[(\mu + \frac{\mu_t}{\sigma_\omega})grad\ (\omega)\right] + \gamma_1\left(2\rho S_{ij} * S_{ij} - \frac{2}{3}\rho \omega \frac{\partial U_i}{\partial x_j}\delta_{ij}\right) - \beta_1 \rho \omega^2 \qquad (20)$$

These equations for the kinetic energy and specific dissipation are described in words in *Equation* (21) below.

$$\begin{aligned} &Rate\ of\ change + Transport\ by\ convection \\ &\qquad = Transport\ by\ turbulent\ diffusion + Rate\ of\ production \\ &\qquad - Rate\ of\ dissipation \end{aligned} \qquad (21)$$

The values of the constants σ_k, σ_ω, γ_1, β_1, and β^* from the transport equations can be seen below in *Table* (2).

Variable	σ_k	σ_ω	γ_1	β_1	β^*
Value	2.0	2.0	0.553	0.075	0.09

Table 2: Table of the constants for the k-ω model transport equations

Near the wall it is not necessary to have wall–damping functions at low Reynolds numbers such as with the k-ε model. Near the wall, the value of k is set to zero and ω is set to infinity. At the inlet, the values of k and ω must be specified and at the outlet a condition of zero gradient is commonly used. In free stream conditions where the kinetic energy and specific dissipation go to zero the model tends to have problems since the eddy viscosity becomes either infinite or indeterminate. To counteract this, a small value greater than zero is chosen for ω, but the results are strongly correlated to this specified value so it cause large inaccuracies.

The k-ω SST (Shear Stress Transport) model

Since the k-ε has been found to have problems near the wall and with adverse pressure gradients, and the k-ω model has problems in the free stream flow, Menter developed the k-ω SST model. This model acts like a k-ω model near the wall and with adverse pressure gradients, and like a k-ε in the free stream. The Reynolds-stresses are solved the same way as the k-ω model, but the equation for the dissipation rate of kinetic energy has ω substituted in for ε [7]. This new transport equation can be seen below in *Equation* (22).

$$\frac{\partial(\rho\omega)}{\partial t} + div(\rho\omega U) = div\left[(\mu + \frac{\mu_t}{\sigma_{\omega,1}})grad\ (\omega)\right] + \gamma_2\left(2\rho S_{ij} * S_{ij} - \frac{2}{3}\rho\omega\frac{\partial U_i}{\partial x_j}\delta_{ij}\right) - \beta_2\rho\omega^2 + 2\frac{\rho}{\sigma_{\omega,2}\omega}\frac{\partial k}{\partial x_k}\frac{\partial\omega}{\partial x_k} \tag{22}$$

It can be seen that this new equations has an extra term. This term represents the cross-diffusion which appears when replacing ε with ω. The model constants are also changed from the k-ε and k-ω models. These new constants can be seen in *Table* (3).

Variable	σ_k	$\sigma_{\omega 1}$	$\sigma_{\omega 2}$	γ_2	β_2	β^*
Value	1.0	2.0	1.17	0.44	0.083	0.09

Table 3: Table of the constants for the k-ω SST model transport equations

Since this model is the combination of two others, where the two models intersect there are some instabilities. To fix this issue, blending functions are used to make a smooth transition between the two. Limiters for the eddy viscosity and turbulent kinetic energy production are used to give improved performance with adverse pressure gradients and wakes. Also, an important difference from the k-ε and k-ω models is that the eddy viscosity is modified to account for the transport of turbulent shear stress in the k-ω SST model [3].

What Turbulence model to use?

As addressed above, the k-ε, k-ω, and k-ω SST models have different strengths and weaknesses depending on the type flow and orientation with respect to the wall. The k-ε model has been found to be good to use in free stream flow conditions, but gets problems once there are adverse pressure gradients of it is used closer to the wall. The k-ω model is more accurate when it comes to adverse pressure gradients and near wall conditions, but it has trouble in free stream flow. The k-ω SST model is a combination of these other two models. In near wall and adverse pressure gradient flow conditions it acts like a k-ω model and in the free stream flow it acts like a k-ε model. Because of this adaptation to different flow conditions, its overall accuracy is much better than either of these other two models. Since the turbines were to be modelled in boundary layer flow near the wall as well as in the free stream, it made sense that a k-ω SST model would be preferable since it could model both of these conditions more accurately. However, we built the numerical model using k-ε but also ran some comparisons to the k-ω SST model.

Actuator Disks

To model the effect of the wind turbine on the flow, a model based on one dimensional linear momentum theory is used. The actuator disk model was used to represent a wind turbine. Actuator disks are a numerical tool that were used to save computational expense when modeling the effect of the wind turbine on the flow. An actuator disk is said to be a "momentum sink" that imparts a force on a flow.

Rather than resolving an entire turbine, actuator disks were used to model the turbines. The momentum of the flow that passes through an actuator disk is decreased corresponding to specified parameters of the actuator disk. In our numerical model, the coefficient of thrust C_t and coefficient of power C_p are input into the model to capture the effect of the turbine on the flow. These input values were measured in an experiment in the FPF. From the front to the back of the actuator disk there is a discontinuity of the pressure. At the front of the disk there is an increase in pressure and this pressure immediately drops below ambient pressure at the back of the disk. The velocity is considered continuous though the actuator disk, but is decreased from the inlet velocity value. In order to use an actuator disk it must be assumed that the flow is incompressible and that the flow is steady through the simulation. Otherwise the flow passing through the front of the actuator disk would be affected by the fluid that had already passed though the disk.

One limitation of an actuator disk is that it does not impart turbulence as a turbine would. This limits an actuator disk's modeling abilities as a wake behind an actuator disk will recover more slowly than a more turbulent wake created by an actual turbine. The turbulence created by the turbine blades will induce mixing in the turbine's wake. This mixing will cause the lower velocity flow inside the wake to more rapidly diffuse outward towards the higher velocity freestream flow outside of the wake. Without this mixing, wakes do not recover nearly fast enough to accurately model real turbines. To correct for this, turbulence sources were added to our numerical model actuator disks. These turbulence sources impart turbulence in the flow directly behind an actuator disk to aid mixing and accelerate wake recovery.

Problem Description

A numerical model representing a wind turbine array was created. In the UNH FPF, wind turbine array studies have been conducted. In this section, the numerical model and physical problem in the FPF are broken down, how they are related, and why we can relate them.

UNH Flow Physics Facility (FPF)

The UNH Flow Physics Facility is the largest boundary layer wind tunnel in the world spanning 72 meters. This great length gives it the ability to produce boundary layers with a height of 1 meter at the back of the tunnel. This facility is currently being utilized to perform experimental tests of wind turbine arrays to observe how the wakes of the front turbines affect the performance of the downstream turbines.

The Physical Problem

The FPF helps keep all the controlled variables required for the studies constant. The assumptions and characteristics of the physical problem that the experiments study in the FPF can be seen below.

Assumptions

- Incompressible flow
- Newtonian fluid
- One dimensional flow
- Zero pressure gradient

Characteristics

- Turbulent boundary layer to model an atmospheric boundary layer
- Model wind turbines (1:500 of a modern 5MW, 125m Diameter wind turbine)
- Porous Disks to represent the model turbines

The Numerical Problem

To build the numerical model, it was crucial to understand the governing equations utilized to model the problem. It was also important to match the boundary conditions to the actual problem. The following are the assumptions and characteristics of the numerical model equations and domain.

Assumptions

- Incompressible flow
- Newtonian fluid
- One dimensional
- Zero pressure gradient in domain

Characteristics

- Boundary layer inlet flow prescribed as a boundary condition
- Actuator disks to represent the model wind turbines and porous disks
- Mesh resolution at and behind the actuator disk

Experiments and Data Acquisition

Numerical models provide a numerical solution to the governing equations. This is only valid if coupled with appropriate boundary and initial conditions and a mesh that describes the geometry of the problem correctly. It was important, as the numerical model was built, that the processes needed to validate the model was understood. A numerical solution is strictly that of a numerical problem and not the physical one. The differences between these two would be caused by inadequate mesh representation of the geometry, a poor choice turbulence model, and wrong boundary conditions. In order to determine whether the numerical model exhibits similar relationships to the physical problem, it was compared to experimental data taken in the UNH FPF. Most of the FPF data utilized for the numerical model was previously acquired data acquired for the many studies done on wind turbine arrays before the start of this project.

Model Wind Turbines

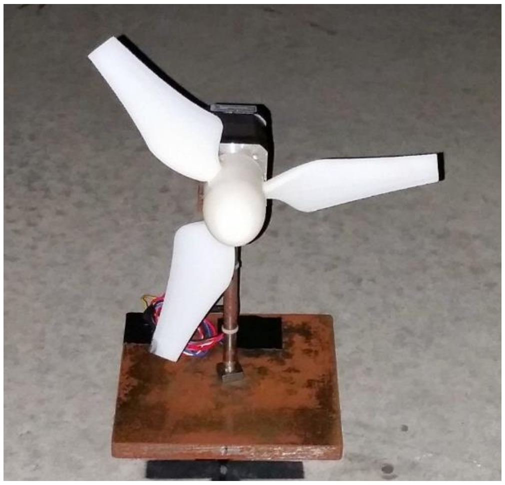


Figure 3: A picture of a model wind turbine taken in UNH FPF

The current model wind turbines utilized for experimentation in the FPF were created by past senior project groups. These turbines have a rotor diameter of D = 0.25 m and their hub height is 0.75D. The blades were designed using NACA 2412 airfoils [2].

Porous Disks

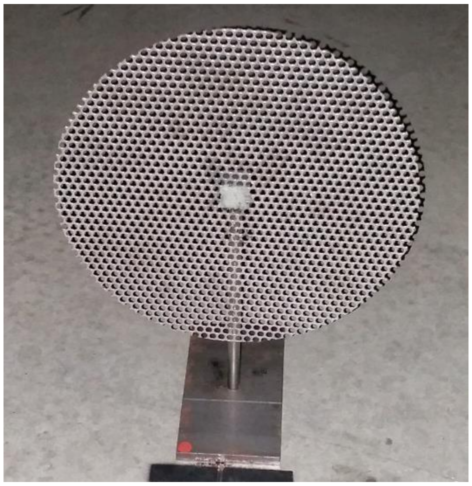


Figure 4: A picture of a porous disk taken in UNH FPF

Porous disks were utilized to represent model turbines in the array studies done in the FPF. Although a porous disk doesn’t affect the flow exactly like the model turbine, its overall effect on the wake at a relevant distance behind it trends very well with that of a model wind turbine. By 8D behind them, the wakes of the model wind turbine and porous disk match very well [2].

Velocity and Turbulence Intensity Profiles

Inlet Profiles

The velocity and turbulence intensity profiles were obtained at the inlet of the FPF at UNH. This was accomplished using a hot-wire on a traverse. Measurements were taken from the bottom surface of the wind tunnel up to one meter high. These inlet profiles were utilized as boundary conditions when modeling flow in the BL. A normalized plot of the inlet velocity profile can be seen in *Figure* (6) [5].

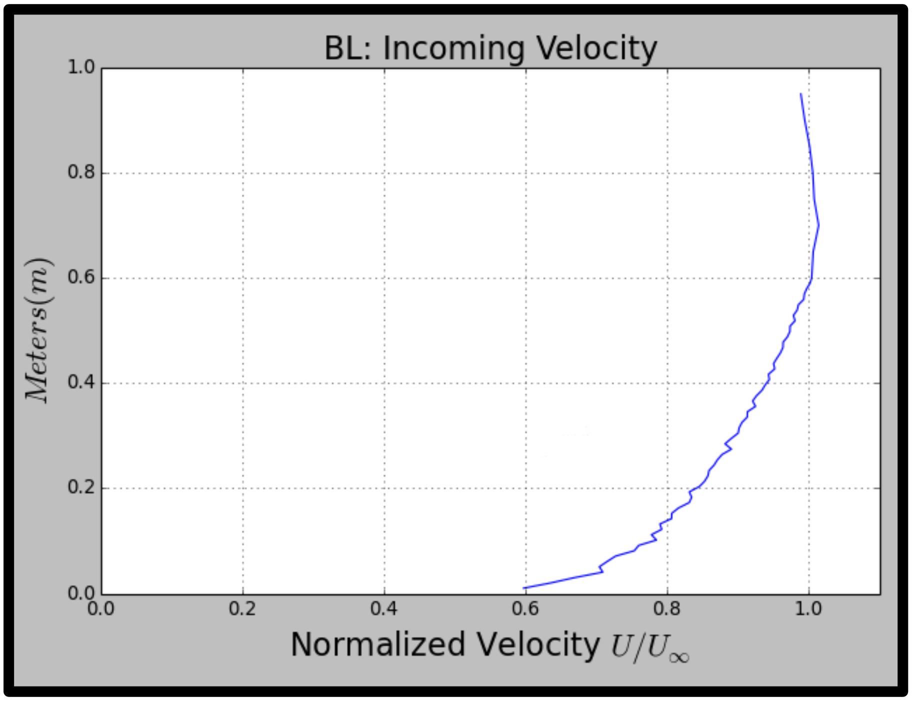


Figure 5: Plot of the inlet velocity from the floor to one meter of the Flow Physics Facility [5]

Profiles behind Model Wind Turbines and Porous Disks

Using a hot-wire attached to a traverse, the velocity profiles behind a single turbine and porous disk were measured at 1D, 2D, 4D, 6D, 8D, 10D, 14D, and 20D behind their location. These measurements were also performed behind the third row model wind turbine of an array. For the velocity profiles, the data was taken for 45 increments starting at hub height to 3D above, vertically. For the turbulence intensity values, the data was taken at 1 cm increments from 1cm above the ground to 1m, vertically. A plot of the velocity profile behind a turbine and porous disk at 8D downstream can be seen in *Figure* (19), and a plot of the velocity profile behind a third row turbine in an array at 20D behind the turbine can be seen in *Figure* (20) [5].

Measurements at Hub Height

Recovery of Velocity Deficit

The velocity was measured behind a porous disk and model wind turbine at hub height using pitot static tube at a frequency of 200 Hz for 10 minutes at each data point. The measurements were taken at 1D, 2D, 4D, 6D, 8D, 10D, 15D, 20D, 25D, and 30D downstream. Each of these data points was then averaged to remove the fluctuations in the velocity value measurements. The velocity deficit was then calculated and plotted. A plot with the recovery of the velocity deficit for the model wind turbine and porous disk together with that of the numerical model can be seen on *Figure* (16) [2].

Downstream Turbulence Intensity

Utilizing a hot-wire at hub height, the turbulence intensity behind a model wind turbine was measured downstream. The turbulence intensity values at 1D, 2D, 6D, and 8D behind the model wind turbine can be seen in Table (4) [5].

Diameters Behind Turbine	Turbulence Intensity Value
1D	0.25
2D	0.22
6D	0.215
8D	0.15

Table 4: Experimental turbulence intensity values at different locations in wake

Coefficient of Power of Turbines in Arrays

To obtain the power coefficient or C_p of the model wind turbines in each row of an array, the turbine stands were secured on load cells to measure the force of the flow being applied to the upstream side of the turbine. From this force measurement, the C_p of the model wind turbines in each row were determined. A plot of the normalized coefficient of power of the model wind turbines for turbine separation of 8D and 10D diameters can be seen on *Figure* (1).

CFD Workstation

Due to the initial difficulty of installing *Ubuntu* and *OpenFOAM* on a computer the group decided early on in the project that it would be worthwhile to build a dedicated computer to run simulations on. The dedicated computer would also allow the group to run simultaneous simulations without having to stop the simulations to perform other tasks.

The work station was designed to be expanded in the future as well as last as long as possible before becoming obsolete. Other considerations taken into account were that the machine should have a large amount of Random-access memory (RAM) to run the memory intensive simulations as well as fairly fast processing capabilities to help solve simulations faster.

An Intel Xeon E5-1630 v3 Quad-Core 3.7 GHz server processor was selected. This processor was selected because the Xeon family of processors are built to operate in servers. They are also built to operate for long durations of time without performing computational errors or crashing. This processor in particular was also built to handle the newest standard of RAM, DDR4.

After selecting the processor, a SuperMicro MBD-X 10SRI-F server mother board was chosen. Initially, this mother board was selected because its LGA 2011 CPU socket accepts the Intel Xeon CPU. However, the mother board was also selected for a multitude of other reasons. Its eight, 284 pin RAM slots can accept DDR4 RAM to hold a total of 512GB of RAM. The mother board also had a multitude of Peripheral Component Interconnect (PCI) slots for a graphics card and 10 SATA ports for hard drives. The SATA ports would also be used in the future to incorporate more processors when expanding to create a small CFD cluster.

After selecting the mother board, RAM was selected. Four sticks of Crucial 8 GB ECC DDR4 server memory were selected. When selecting RAM, the newest standard, DDR4 RAM was chosen to prevent the machine from becoming obsolete. In addition to future proofing the machine, this RAM operates with less latency at the value of 15. Latency is a measure of the time between receiving a command and performing an operation and less latency means a better performance. Four sticks of RAM were chosen such that each core of the processor would have its own dedicated RAM to operate more efficiently. The RAM is also enabled with ECC capabilities. ECC stands for Error-correcting code memory. ECC RAM has the ability to detect and correct common forms of internal data corruption. This is necessary when running large simulations that take long amounts of time to run. If a section of the memory became corrupted, the simulation would crash resulting in large amounts of time being lost between when the failure occurred and when it was realized and restarted by the operator.

A Samsung 2.5” 120 GB solid state internal hard drive was selected to run the operating system and *OpenFOAM* off of. A solid state hard drive operates at much lower latency than a typical optical hard drive and will allow *OpenFOAM* and the operating system to perform faster than if they were run off an optical hard drive.

A 3.5” 1 TB Western Digital optical hard drive was also purchased to store simulation results on. The speed of this hard drive was not as crucial so an optical hard drive was purchased due to its affordability.

The entire work station runs off a 500 Watt Corsair power supply and is encased in a 4U rackmount server case. Additional cores of the workstation would run off their own power supply and the rackmount server case allows for easy stacking of cores for when the workstation is built out.

Surplus monitors, keyboard, and mouse were used to save expense. A full list of all components used in the machine are given in *Table* (5) below. A picture of the workstation assembly can be seen in *Figure* (7).

Item	**NewEgg PN (If App)**	**QTY**	**Price**	**Description/Source**	
1		1	$371.86	Intel Xeon Processor	**Total** $1,325.76
2	N82E16813182928	1	$287.99	SuperMicro Mother Board	
3	N82E16820148842	1	$109.99	Crucial 32GB(4X8GB) RAM	
4	N82E16820147247	1	$79.99	Samsung 120GB 2.5" SDD	
5	9SIA30R2AV9542	1	$154.99	WD 1TB 3.5" HDD	
7	N82E16817139107	1	$29.99	500 W Corsair Power Supply	
8	N82E16811182566	1	89.99	Rackmount Server Case	
9	N82E16814105002	1	$119.99	AMD Graphics Card	
10	N82E16816101683	1	$36.99	SuperMicro CPU Heatsink	
11		1	$35.99	Wifi Card	
12	N82E16817997013	1	$7.99	2.5" to 3.5" HD Adapter	

Table 5: Computer component costs

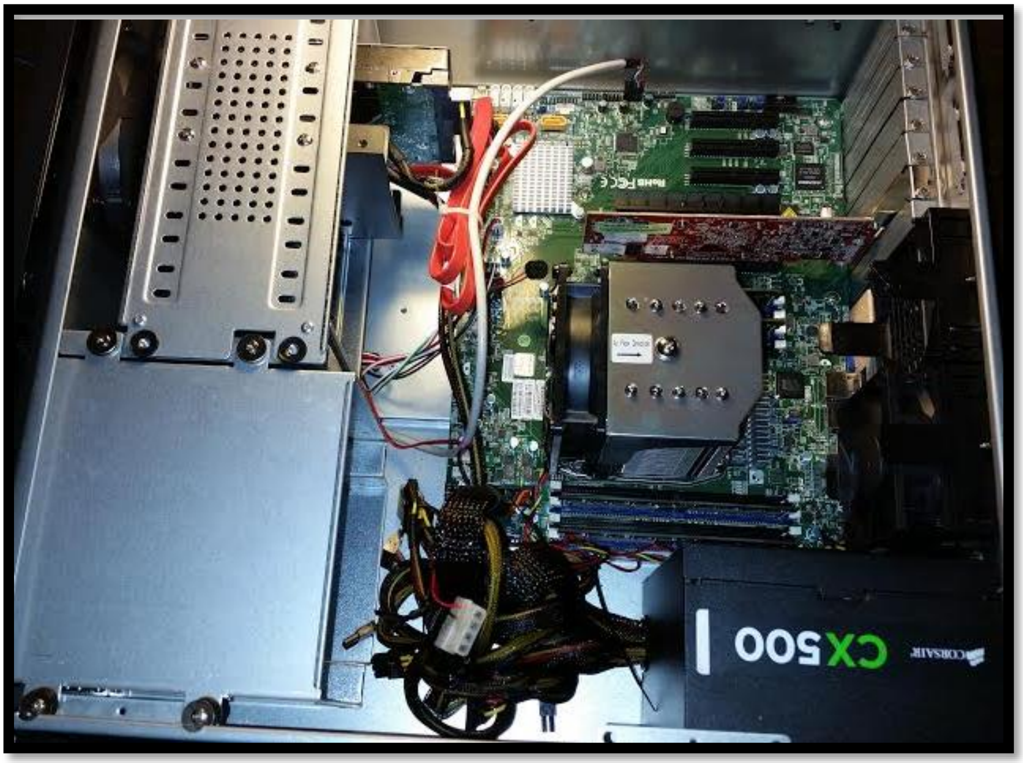


Figure 7: A picture of the CFD Workstation

The new workstation solves cases approximately 4 times faster than the 4 core, Intel i5 laptop that was previously being used to run the majority of simulations. It also will allow future CFD design groups to start up faster as they will not have to work through the process of setting up their computers to dual boot into *Ubuntu* and then install *OpenFOAM* on their own computers.

Open Sourced Engineering

The Operating System (OS) that was utilized for our CFD workstation was *Ubuntu* because it was necessary for us to use a Linux based OS to run the software needed for the numerical model simulations. To build our numerical model, *OpenFOAM*, a C++ based CFD software was used. *Paraview* is the data viewing software that is paired with *OpenFOAM*, and this was used for visualization and post-processing of the results from the numerical model. *Python* was also utilized to do post-processing of the numerical simulation data. Its powerful characteristics were utilized to sort data and create plots. To keep track of the code changes, the version control system *Git* was used. The source code sharing and hosting site, *Github* was utilized to share our numerical model code.

Numerical Model in OpenFOAM

OpenFOAM stands for Open Source Field Operation and Manipulation. It is a C++ based program that is applied to solving ordinary as well as partial differential equations for many different fluid based applications. Since it is open source, it is free and constantly under development, but it has the same capabilities as many commercial computational fluid dynamics applications. It is used widely in industry, academia and in research labs. It is a finite volume method based solver and uses polyhedral unstructured meshes. Finite volume method is the methodology utilized by OpenFOAM and our numerical model. It is also known as a control volume method as it is based off of integration on control volumes of the domains [5].

Actuator Disk

In the numerical model, wind turbines were represented as actuator disks. The coefficient of power C_p and coefficient of thrust C_t were input into the code from data taken in the FPF. Relationships known between the axial induction factor and the flow are used to determine the effect of the turbine on the flow through the code. The reason why actuator disks were chosen was their simplicity and reduction on the required computational power.

Turbulence Model Choice

Our selected choices of RANS turbulence models for the numerical model were the k-ε and the k-ω SST two equation models. We used these two models knowing their specific strengths weaknesses and also wanted to determine which would best be used for our numerical model. We ran simulations utilizing both models but did our final runs with the k-ε model.

OpenFOAM Solver

To build the numerical model, the *simpleFoam* utility in OpenFOAM was used. This utility is a steady state solver for incompressible and turbulent flow. This solver together with running the case in parallel were utilized to solve the numerical problem. Running in parallel required to spread the model domain into four processors of the CFD workstation.

SIMPLE Solution Algorithm

The SIMPLE algorithm, also known as the Semi-Implicit Method of Pressure Linked Equations was the solution algorithm utilized in our OpenFOAM numerical model. This is an algorithm that utilizes a guess and correct procedure to calculate the pressure and velocity in the flow in a grid arranged in a staggered alignment. It can be seen on *Figure* (8) below the procedure of a CFD solution using the SIMPLE algorithm in a 2-D flow. The turbulence model equations are discretized and solved to provide us with solutions for our model [5].

Parameter	Description
p^*	Guessed pressure field
u^*, v^*	Velocity components from guessed pressure field
ϕ^*	Other transport equation variable. In our case k and ε

Table 6: Some parameters in the SIMPLE algorithm

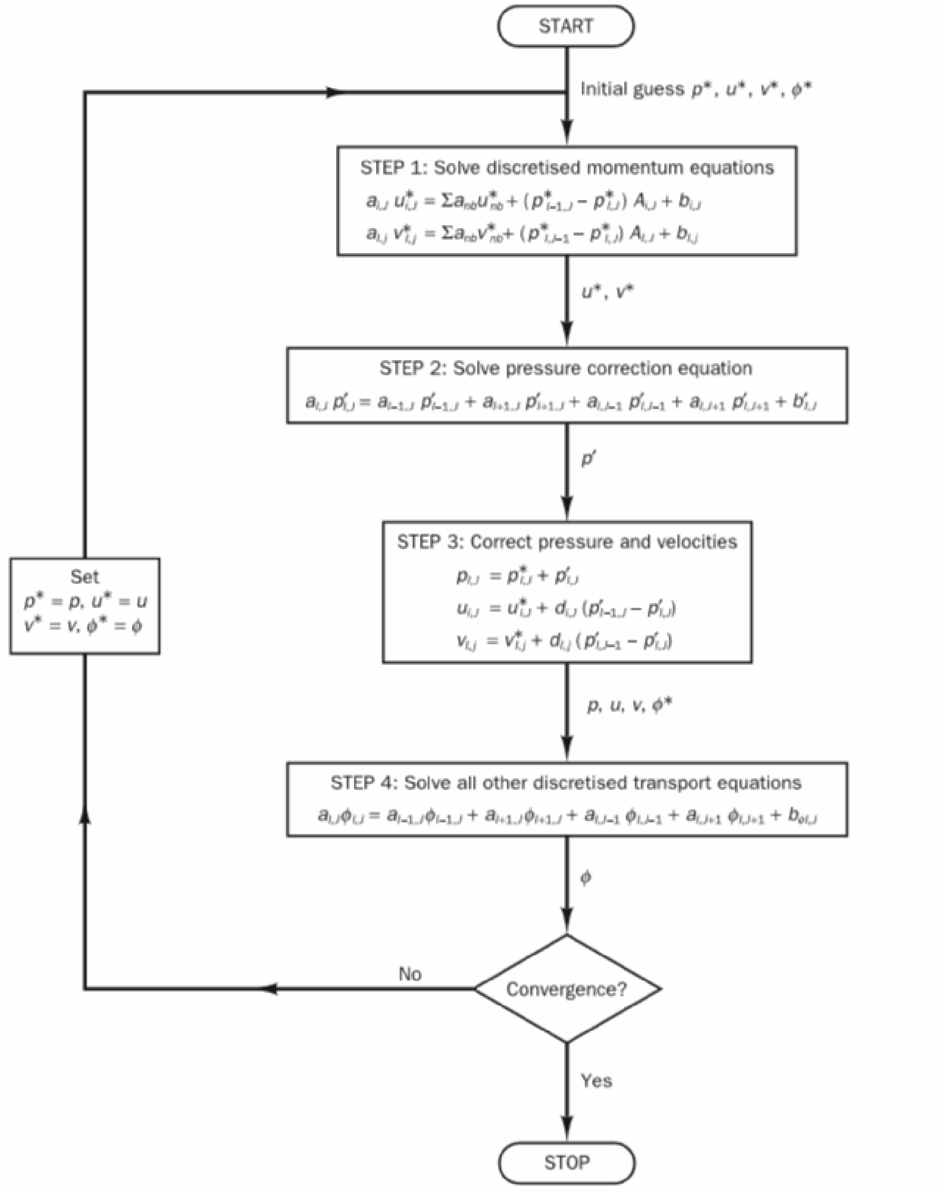

START
Initial guess p^*, u^*, v^*, ϕ^*
STEP 1: Solve discretised momentum equations
$a_{i,J}\, u^*_{i,J} = \Sigma a_{nb} u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J})\, A_{i,J} + b_{i,J}$
$a_{I,j}\, v^*_{I,j} = \Sigma a_{nb} v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J})\, A_{I,j} + b_{I,j}$
u^*, v^*
STEP 2: Solve pressure correction equation
$a_{I,J}\, p'_{I,J} = a_{I-1,J}\, p'_{I-1,J} + a_{I+1,J}\, p'_{I+1,J} + a_{I,J-1}\, p'_{I,J-1} + a_{I,J+1}\, p'_{I,J+1} + b'_{I,J}$
p'
STEP 3: Correct pressure and velocities
$p_{I,J} = p^*_{I,J} + p'_{I,J}$
$u_{i,J} = u^*_{i,J} + d_{i,J}\,(p'_{I-1,J} - p'_{I,J})$
$v_{I,j} = v^*_{I,j} + d_{I,j}\,(p'_{I,J-1} - p'_{I,J})$
p, u, v, ϕ^*
STEP 4: Solve all other discretised transport equations
$a_{I,J}\phi_{I,J} = a_{I-1,J}\phi_{I-1,J} + a_{I+1,J}\phi_{I+1,J} + a_{I,J-1}\,\phi_{I,J-1} + a_{I,J+1}\,\phi_{I,J+1} + b_{\phi I,J}$
ϕ
Convergence?
No
Yes
STOP
Set
$p^* = p$, $u^* = u$
$v^* = v$, $\phi^* = \phi$

Figure 8: Sequence of operations and procedures that utilizes the SIMPLE algorithm [5]

Boundary Conditions (k- ε turbulence model)

This is one of the most important parts of the numerical model. Selecting the correct boundary conditions to utilize for the model is essential in the validity of a CFD model. Our model is a steady state model whereby iterations lead to the final, converged flow field. The model also assumes a zero pressure gradient internal field meaning there isn't a change in the pressure throughout the domain. In order to obtain the turbulence intensity I and length scale l to calculate boundary conditions in the freestream, it was necessary to resolve the entire wind tunnel up until near where the wind turbines were located during the experiments. By doing this it was possible to get accurate boundary conditions right in front of the actuator disk to allow shortening of the model domain and save on computational power for the final model. The turbulence intensity in the inlet of the wind tunnel is 0.5% and the length scale was measured at the inlet screens and determined to be 1.5 mm. A diagram of the domain utilized for the final simulations can be seen in Figure (9).

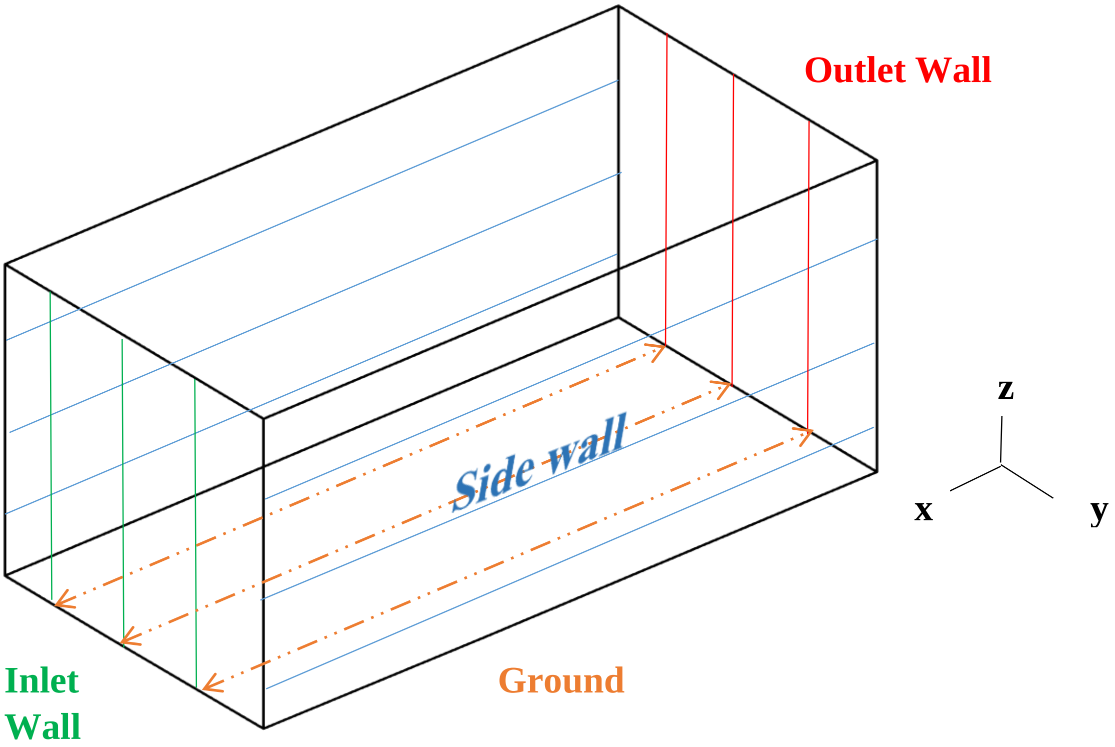


Figure 9: Domain setup in OpenFOAM. Flow is 1-D in the x direction. The z direction is upwards. An actuator disk was placed in the domain.

Parameters	Upper Wall and Side Wall (FS & BL)	Outlet (FS & BL)	Ground (FS & BL)
p	$slip$	$internalfield$	$slip$
U	$slip$	$internalfield$	$slip$
k	$slip$	$internalfield$	$slip$
ε	$slip$	$internalfield$	$slip$

Table 7: Boundary Conditions of the domain

Parameter	**Description**	**FS (Tunnel Inlet)**	**FS (Domain Inlet)**	**BL (Domain Inlet)**	**Units**
U	*Mean velocity*	6.82	6.82	**varying**	$\frac{m}{s}$
I	*Turbulence intensity*	0.5	**None**	**varying**	%
k	*Turbulence kinetic energy*	$\frac{3}{2}(UI)^2$	0.000165	**varying**	$\frac{m^2}{s^2}$
l	*Length scale*	0.0015	**None**	0.0045	m
ε	*Kinetic energy dissipation*	$\frac{C_\mu k^{\frac{3}{2}}}{l}$	$4.46e-5$	**varying**	$\frac{m^2}{s^3}$

Table 8: Inlet Boundary Conditions. The FS Domain Inlet BC were resolved from the FS full tunnel simulation.

Inlet boundary condition in the boundary layer was prescribed from data in the UNH FPF *Figure* (6). *OpenFOAM* code was adapted to insert the boundary layer flow, eliminating the need to resolve one due to lack of computational power for a large, high resolution domain. Measured velocity and turbulence intensity inlet profiles were used to create the varying values prescribed on the inlet boundary condition to represent boundary layer flow. A constant length scale value was calculated from the k and ε values resolved for at the domain inlet using *Equation* 24. A diagram of the interpolation across the inlet surface to apply the boundary layer flow from the FPF can be seen in *Figure* (10).

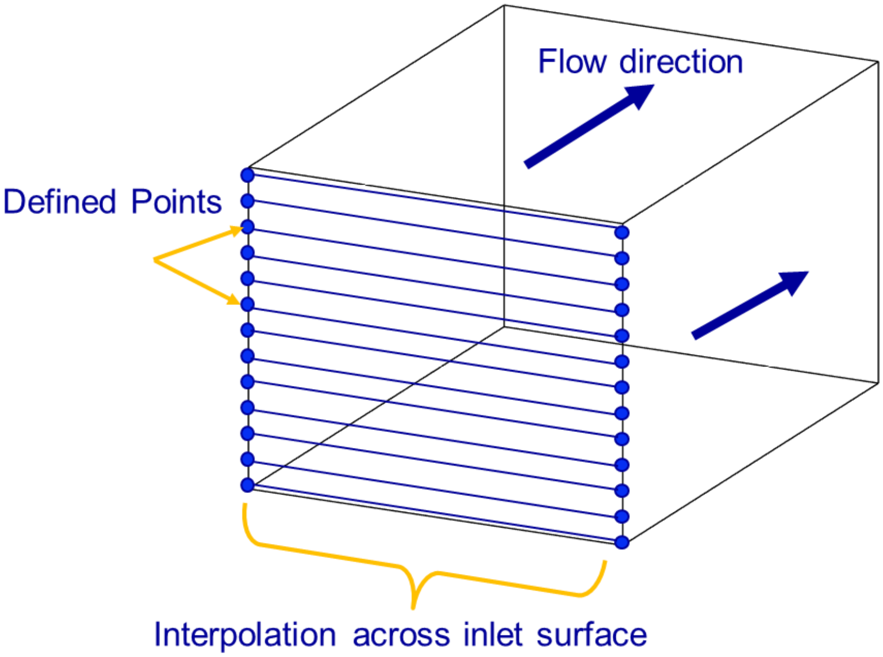


Figure 10: The UNH FPF boundary layer, BC data interpolated across the points of the domain for our BC

Mesh

A great amount of time on this project was spent developing a mesh for the numerical model. The domain or the mesh of the model was a very crucial part of our work. Depending on the computer computational capabilities or the detail needed want to resolve the model, the mesh size selected can be the main limitation of the model. Varying the mesh size can also play a large role on the solution and whether or not it converges to a solution that makes sense. This is something that had to be studied in order to keep track of the validity of our numerical model. Mesh refinement was performed using the *OpenFOAM* utility known as *snappyHexMesh*. Areas of interest were chosen to refine the mesh and these were varied for each validation step taken. Main areas of high refinement were the actuator disk area and the wake behind the turbine where high resolution is crucial to capture the detail in the flow. Pictures generated using *Paraview* show the mesh of the models in the freestream and boundary layer in *Figure*s (11) and (12) respectively.

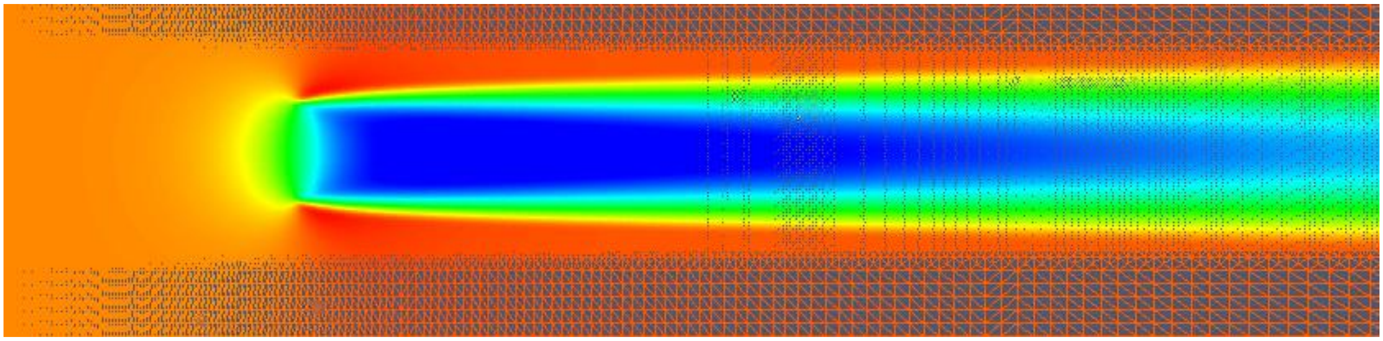


Figure 11: A 2-D slice showing an actuator disk in the freestream with a refined mesh. The mesh refinement decreases as we move further away from the actuator disk.

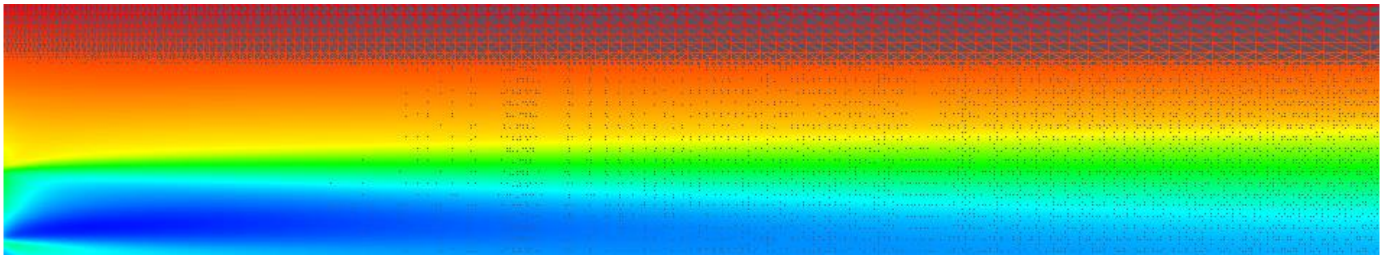


Figure 12: A 2-D slice showing an actuator disk in the boundary layer with a refined mesh. The mesh refinement decreases as we move further away from the actuator disk and also as we move further away from the bottom.

Post Processing

After the numerical model had converged, the data was analyzed to interpret the results. This was first done by visualizing the data in *Paraview*. From *Paraview,* the velocity, turbulence kinetic energy, dissipation of turbulence kinetic energy, and other parameters could be viewed throughout the domain. To obtain plots of the results and to plot alongside experimental data, *Python* was used. These plots generated using *Python* were used to determine any similar trends between the numerical models and experimental data.

Results and Discussion

Numerical Solutions and Validation

In order to make sure we made our numerical model correctly, we took small steps toward building it into a turbine array. After each step, the simulated results were validated by comparing them to experimental data. The first step was to have a simple flow passing through an empty domain. The next step was then to put an actuator disk into a uniform free stream flow. Once this step was validated, a boundary layer flow was then applied to the inlet of the domain. After verification of this result, two more actuator disks were added to the domain to make a 3 by 1 array. Data from behind the third row actuator disk was then compared to that of a third row model wind turbine in an experimental array. We used the k-ε turbulence model but did run some initial simulations in k-ω SST.

Single Actuator Disk Freestream Flow

We used both the k-ε and k-ω SST models to simulate a simple, uniform flow in a domain. Once we were able to apply boundary and initial conditions to a domain with a simple flow, an actuator disk was applied to the domain with a uniform free stream inlet flow. While looking at the velocity in the wake of the disk, it was found that the velocity recovery occurred too slowly to match with an actual model wind turbine or porous disk. This was the case for both the k-ε and k-ω SST turbulence models. To determine why this was, the turbulence kinetic energy was observed. It was then realized that the only turbulence being generated behind the disk was due to the shear layers at the boundaries of the actuator disk wake due to the gradient of the velocity field. A picture of the turbulence kinetic energy being generated by the velocity gradient can be seen on *Figure* (13).

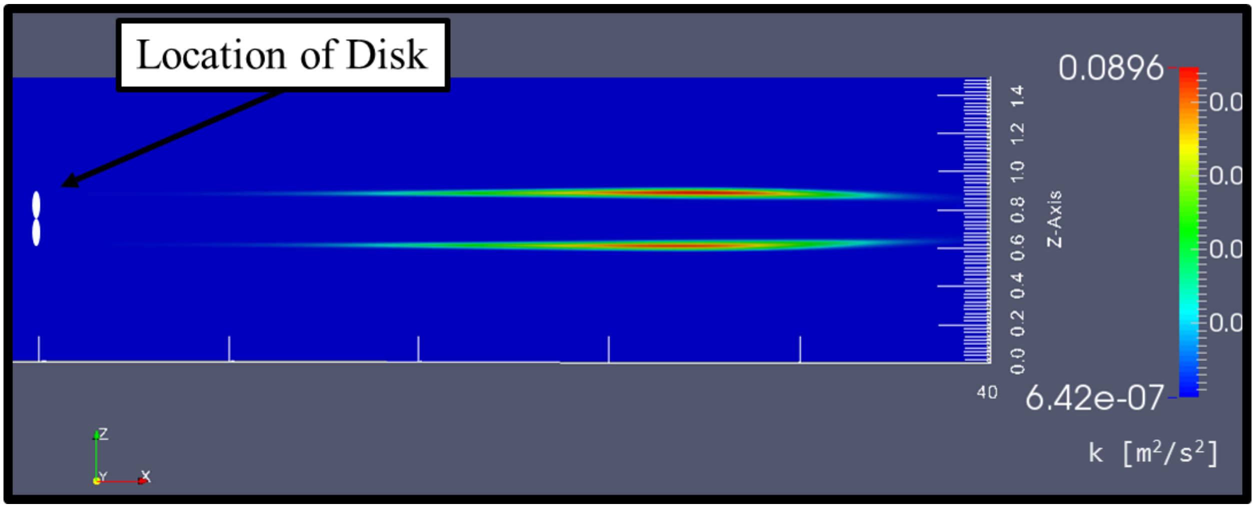


Figure 13: A 2-D Paraview visualization of an actuator disk in a Freestream flow using the k-ε turbulence model plotting the k distribution. Here we see the lack of turbulence and thus the turbulence kinetic energy k at the actuator disk and the presence only at the shear layers where there exists a velocity gradient.

The reason this was the only turbulence kinetic energy observed was because the actuator disk does not resolve the hub and blades that exist in an actual wind turbine. The presence of these parts would cause mixing of the flow and increase the turbulence and thus the turbulence kinetic energy at the turbine location. To compensate for the absence of the hub and blades, it was determined that turbulence kinetic energy k and turbulence kinetic energy dissipation ε needed to be injected at the actuator disk to represent the lack of turbulence generation from an actual wind turbine. With the right amount of turbulence generation in the form of k and ε, the velocity recovery should occur at a reasonable rate.

This required us to modify the k-ε and k-ω SST turbulence model code in *OpenFOAM*, a significant time consuming feat. We were able to access code for the *OpenFOAM* k-ε model that could inject k and ε values at a specified location. This code was developed by a UNH PhD student and our Graduate Student Advisor, Pete Bachant. Unfortunately, we were not able to adapt similar code to the k-ω SST model and thus had to stop working with it in the building of the numerical model.

The values for the k and ε injection were determined from an experimentally obtained turbulence intensity value 1D behind a model wind turbine, and using the inlet velocity U_o going into the actuator disk. The calculations for the values can be seen in *Equation* (23) and *Equation* (24)

$$k = \frac{3}{2}(U_o I)^2 \tag{23}$$

$$\varepsilon = \frac{C_\mu k^{\frac{3}{2}}}{l} \tag{24}$$

where U_o is the inlet velocity to the actuator disk and I is the turbulence intensity obtained experimentally. The turbulence injection would be done uniformly along the surface of the actuator disk. A picture of the model with the turbulence injection can be seen in *Figure* (14).

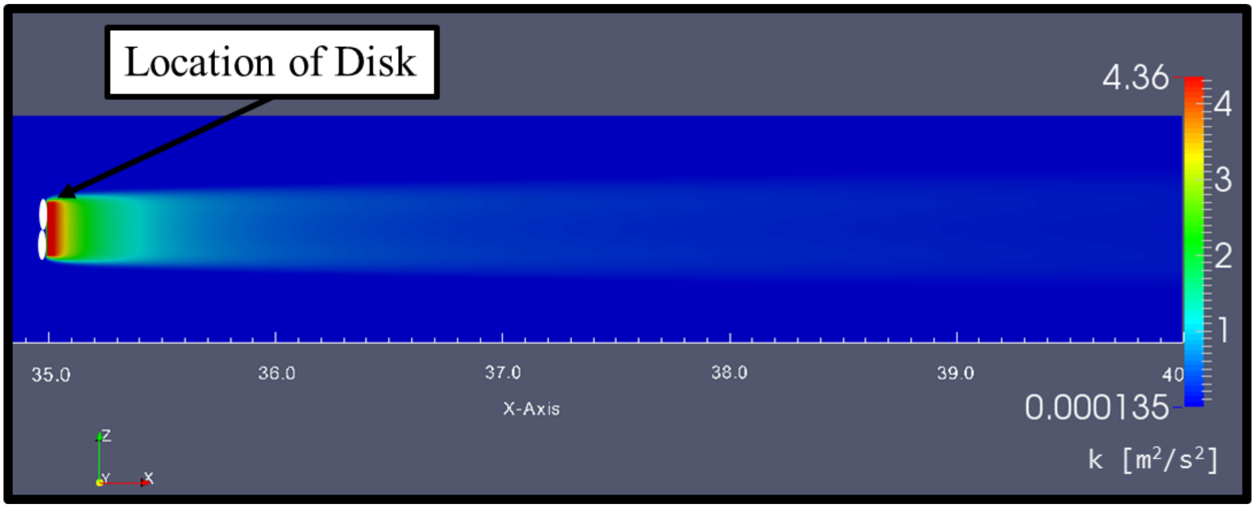


Figure 14: A 2-D Paraview visualization of an actuator disk in a Freestream flow plotting the k distribution. Note the turbulence at the actuator disk represented by the high k values.

Now with the turbulence injection at the actuator disk through k and ε, the velocity results of the simulation were observed and compared to experimental data. A picture of the velocity magnitude of this model can be seen on *Figure* (15).

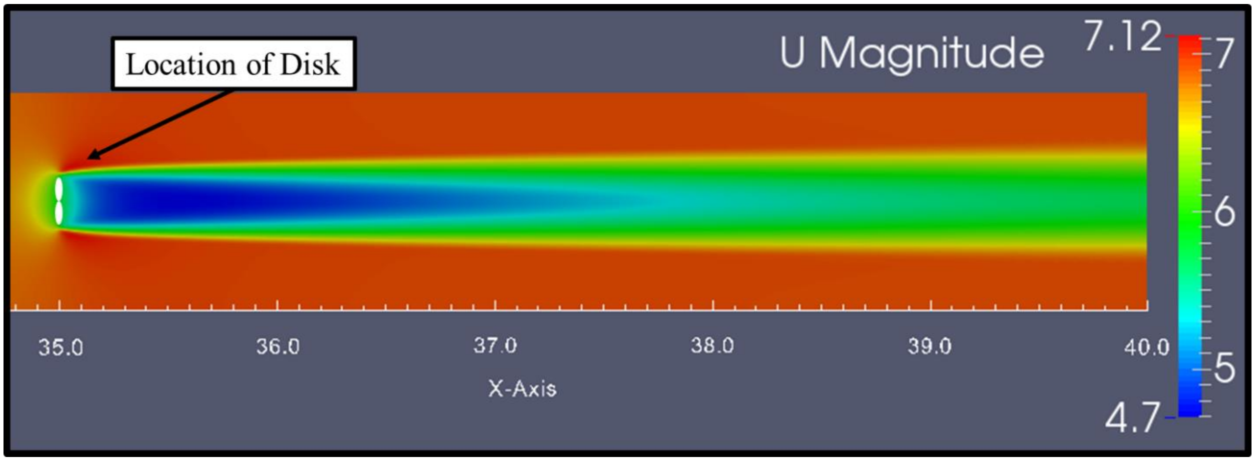


Figure 15: A Paraview visualization of an actuator disk in a Freestream flow plotting the mean velocity distribution. Here we see the wake recovery and mean velocity distribution in a 2-D cross section cutting through the actuator disk.

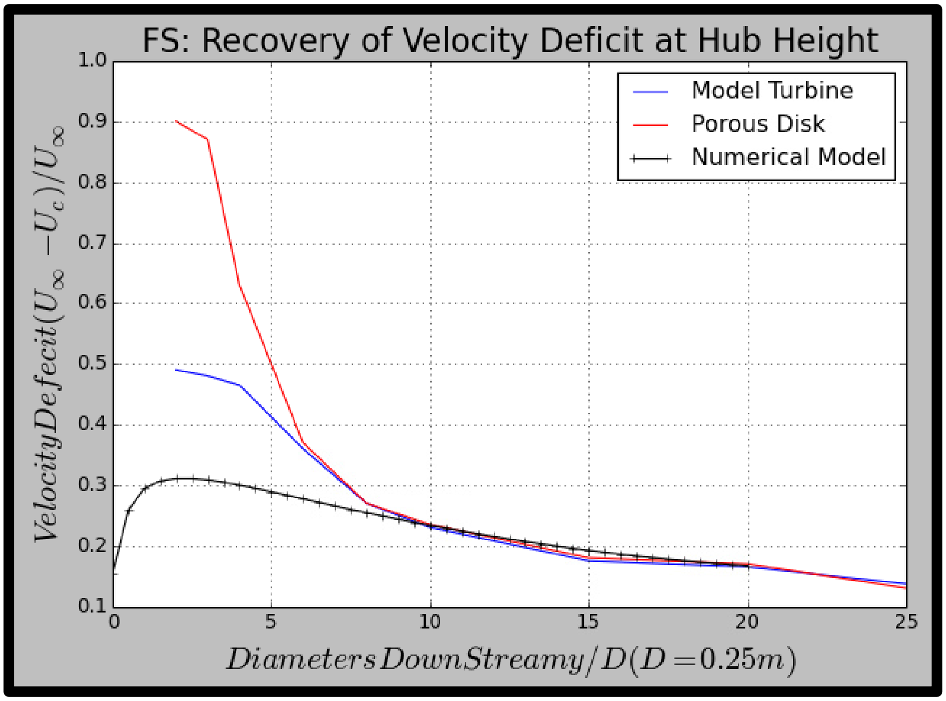


Figure 16: Velocity deficit at hub height for a model wind turbine, porous disk and numerical model in the Freestream. Also note that the model wind turbine and porous disk velocity deficit trend well and are similar by about 8D downstream.

The velocity deficit behind the actuator disk was compared to that of a model wind turbine and porous disk. A plot of this comparison can be seen on *Figure* (16). From this plot, it can be seen that the numerical model matches the velocity deficit of the turbine and porous disk by approximately 8D downstream from the actuator disk. It also fits the trend of the turbine and porous disk very well after 8D. It can also be seen that right behind the actuator disk, the velocity deficit is much less than that of the turbine and porous disk. This is because the actuator disk is only a momentum sink capturing the effect of a wind turbine and is not a physical barrier blocking the fluid flow. For this numerical model, the trends further downstream in the flow by at least 8D are the most crucial. Therefore, this difference between the experimental and simulated results can be ignored.

Single Actuator Disk Boundary Layer Flow

When the model of a single actuator disk in a freestream flow was verified with experimental data, a boundary layer flow was then applied to the inlet of the domain. This boundary layer flow data was obtained from measurements at the inlet of the FPF. These measurements were then input as an inlet boundary condition into domain so the model has an exact representation of the boundary layer inlet flow for the wind tunnel. The numerical model with the boundary layer flow can be seen on *Figure* (17).

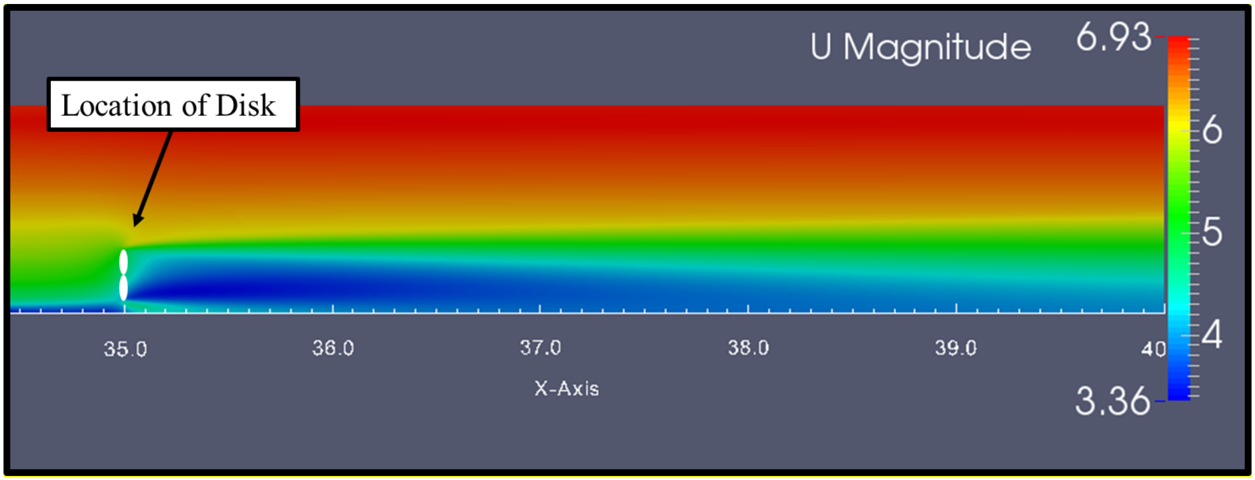


Figure 17: A Paraview visualization of an actuator disk in a boundary layer flow. Here we see the wake recovery and mean velocity distribution in a 2-D cross section cutting through the actuator disk. Note that the velocity doesn't.

The same comparison to experimental data was performed as with the velocity deficit in the freestream flow model. In the boundary layer flow model, it was again matched at approximately 8D behind the actuator disk. It also had the lower velocity deficit value right behind the actuator disk. The flow further than 8D in the wake matches the deficit of the turbine and porous disk very well. This can be seen in the velocity deficit plot in *Figure* (18) below.

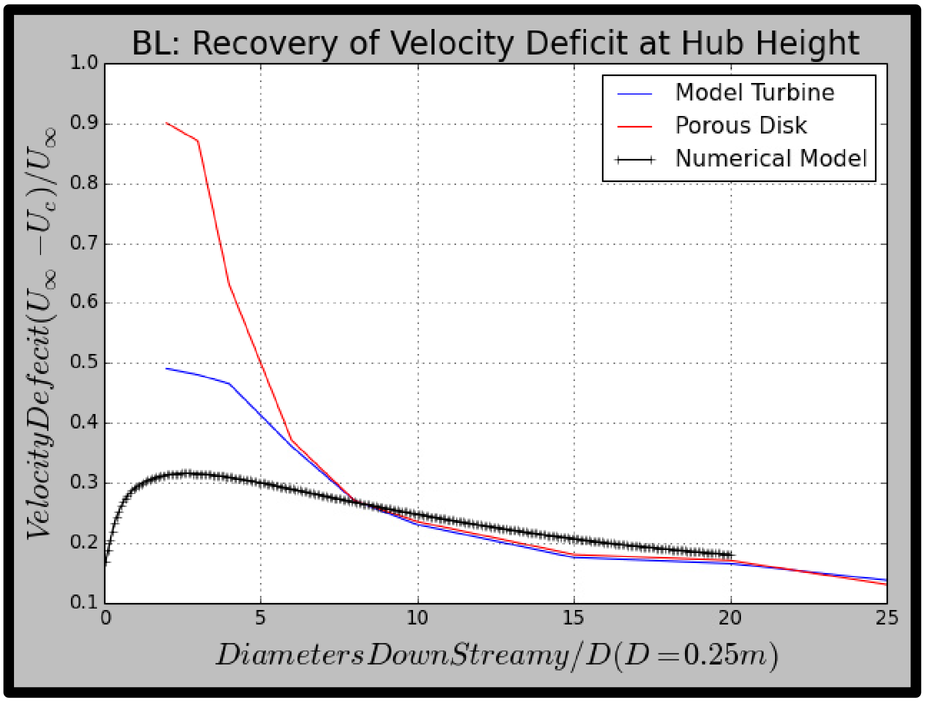


Figure 18: Velocity deficit at hub height for a model wind turbine in the Boundary Layer

Along with a velocity deficit comparison, a velocity profile comparison at distance downstream in the wake was performed. Examples of these profile comparisons at 8D and 20D downstream can be seen on *Figure* (19) and *Figure* (20) below.

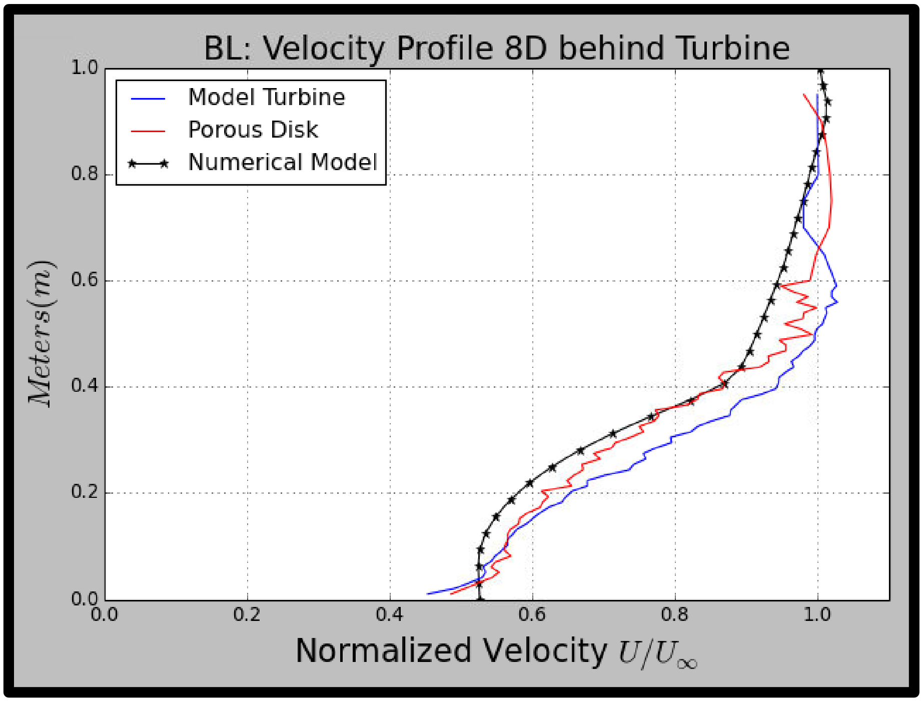


Figure 19: Normalized velocity at 8D behind model wind turbine. Hub height is at 0.1875m or 0.75D.

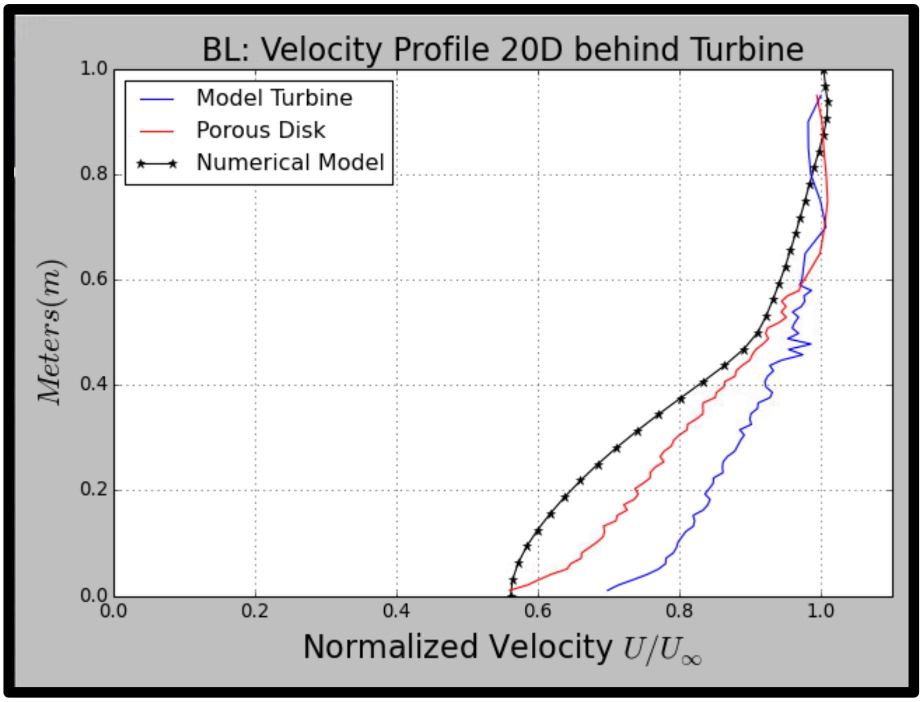


Figure 20: Normalized velocity at 20D behind model wind turbine. Hub height is at 0.1875m or 0.75D.

From the plot shown on *Figure* (19), it can be seen that the velocity profile trend of the numerical model matched that of the experimental data very well in terms of the overall trend of the data. However, major differences can be observed near the wall starting at about 0.1 m to t ground.

From the plot shown on *Figure* (20), it can be seen that the velocity profile trend of the numerical model matched well until at about 0.5m to the ground. Major differences can be observed near the wall, below 0.5 m from the ground. It can be seen that the trend of the numerical model's velocity profile in the far wake at 20D did not match as well as that closer to the actuator disk at 8D.

At the bottom of both of these plots, where the height of the flow approaches zero, it can be seen that there is a difference in the way the flow of the numerical model acts compared to that of the experimental data. It does not approach zero as the experimental data for the turbine and porous disk do. This difference between the modeled and experimental results is due to the use of a slip boundary condition for the ground in the numerical model.

The reason why this was done is due to the type of model used for the simulation. We used a k-ε model which is known to have less accuracy in near wall flows and needs extremely high mesh resolution near the wall for its wall functions to work properly. Since the mesh resolution near the wall with this model was not very high, the accuracy in this area was limited while using this type of model. When the wall boundary condition was set to no slip, a boundary layer flow governed by wall functions began to grow within the inlet boundary layer flow that was already prescribed as a boundary condition at the inlet of the domain. An example of the numerical model solution when using a no slip boundary condition can be seen on *Figure* (21). Due to this issue, we had to reapproach the continuation of building the numerical model.

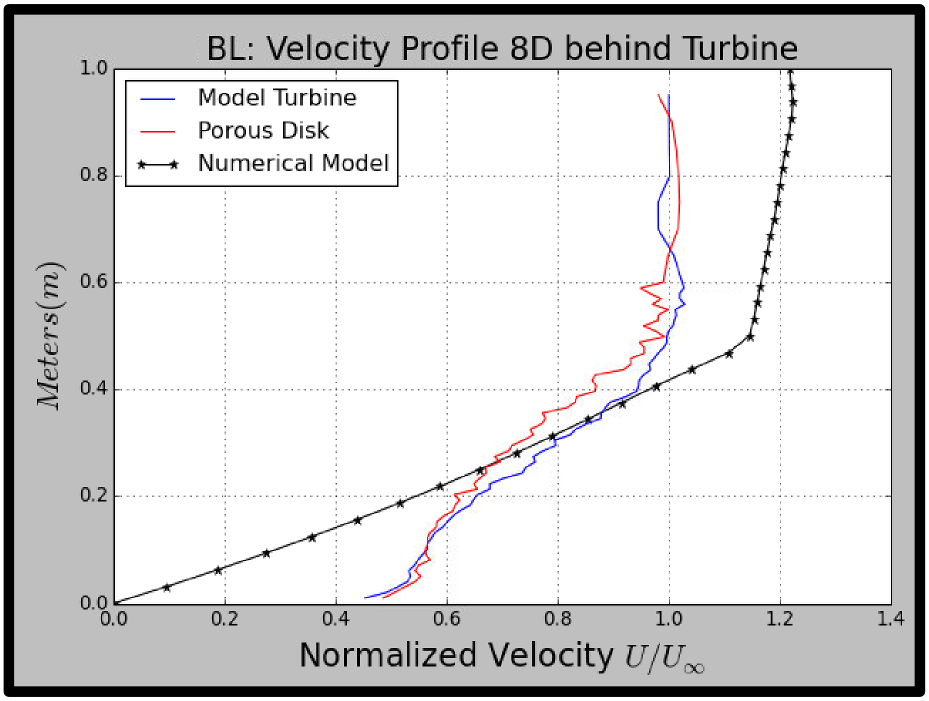


Figure 21: The results at 8D with a slip boundary condition which activates wall functions to determine the velocity distribution. Due to lack of high enough resolution which needs to be at a Y+ of less than 5, the wall functions fail to capture the correct distribution.

In order to continue with building the numerical model, it was justifiable to use the slip boundary condition, even though it doesn't represent the actual no slip condition on the wall. What this resulted to was that the velocity at the wall was non-zero and would be accelerated by the flow field above it as you move further into the domain. Note that on *Figure* (6), *Figure* (19) and *Figure* (20), the experimental data does not go all the way to the ground or at zero due to data acquisition limitations. We wanted to observe whether or not the model would capture the right trends starting from at least hub height and above. We didn't need the details in the near wall region since we had already prescribed the flow in the boundary layer through the inlet boundary condition. A no slip boundary condition would have resulted in a zero value on the ground but would have caused the model to resolve another boundary layer. If we were able to achieve higher domain refinement, we could have continued with the no slip boundary condition to study its effects on the results.

Actuator Disk Arrays in Boundary Layer Flow

After verifying the numerical model trends for a single actuator disk in a boundary layer flow, the array numerical model was then set up. This was done by adding two more actuator disks to the domain spaced 8D apart to make a 3 by 1 array in the boundary layer. The coefficient of power values for the two back turbines were determined from normalized coefficient of power values obtained experimentally from turbine arrays. From these coefficient of power values, the turbulence kinetic energy injection values for these actuator disks were then calculated. This can be seen in *Equation* (24)

$$k = C\left(\frac{1}{2}U_o^2 C_p\right) \tag{24}$$

where $C = 0.55147$ is a constant determined from comparing the turbulence kinetic energy to the turbulence intensity and the coefficient of power, U_0 is the average velocity entering the region of the actuator disk, and C_p is the coefficient of power of the disk. An illustration of this 3 by 1 array can be seen in *Figure* (22).

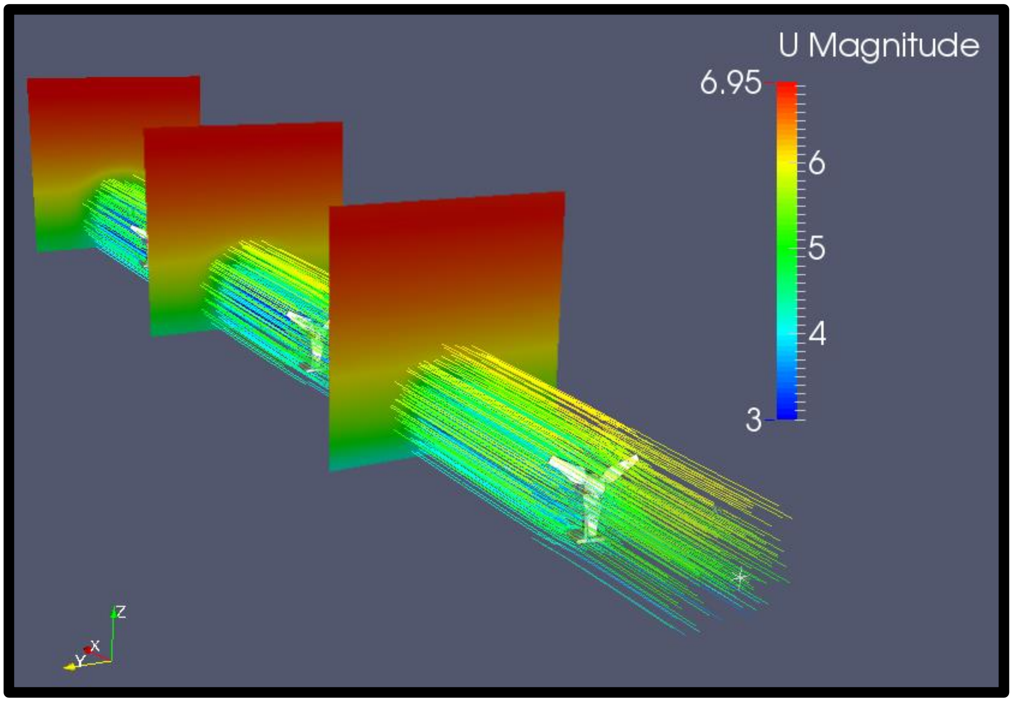


Figure 22: Illustration of the numerical model with a 3 by 1 array

The numerical model results were compared to experimental data by looking at the velocity profiles along the wake of the third row actuator disk in the array and comparing it to velocity profiles behind the third row turbine of an array in the Flow Physics Facility. A plot of the comparison between the experimental and simulated results at 2D behind the actuator disk and model wind turbine can be seen in *Figure* (23).

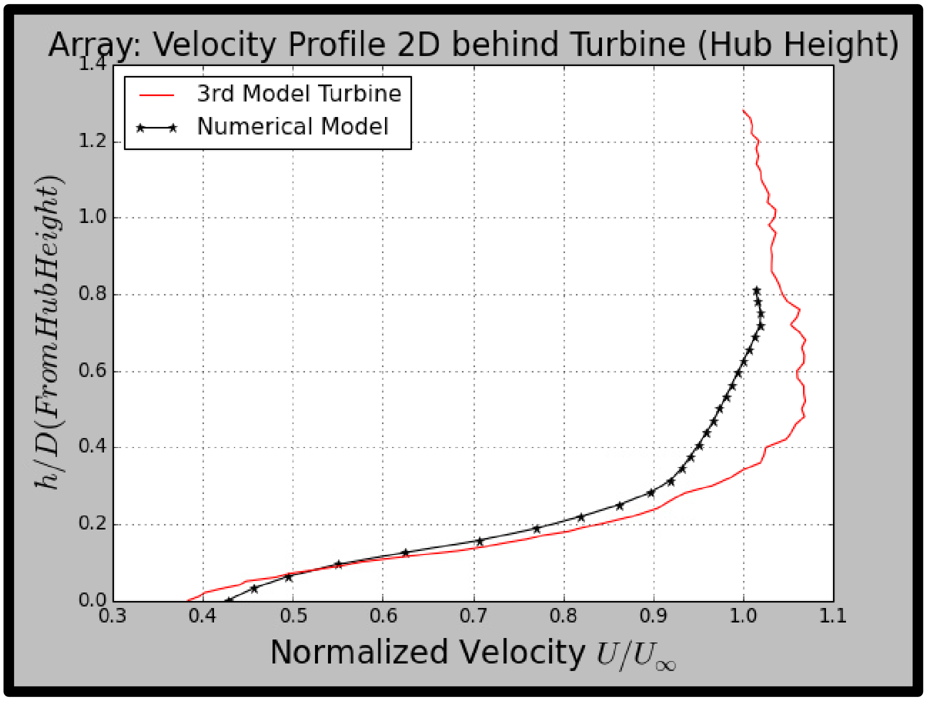


Figure 23: Normalized velocity profile at 2D behind 3rd turbine and actuator disk starting at hub height.

From the plot in *Figure* (23) showing the comparison between the simulated and experimental results behind the third model wind turbine and actuator disk at 2D in the wake it can be seen that the trend of the numerical model results matches the experimental trend very well. Another comparison was made at 4D in the wake. A plot of this comparison can be seen on *Figure* (24).

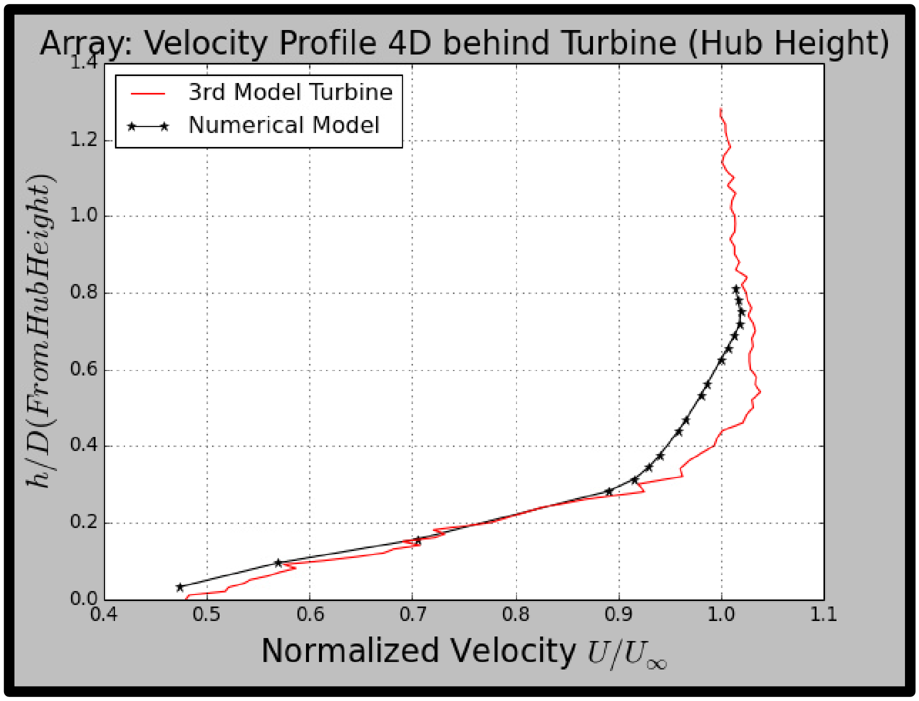


Figure 24: Normalized velocity profile at 4D behind 3rd turbine and actuator disk starting at hub height

The plot shown in *Figure* (24) shows that the trend of the simulated results match the trend of the experimental very well at 4D downstream from the model wind turbine and actuator disk. To obtain a quantitative measure of the accuracy of the numerical model, a plot of percent difference between the simulated and experimental profiles. The plot of the percent difference at 2D in the wake can be seen in *Figure* (25).

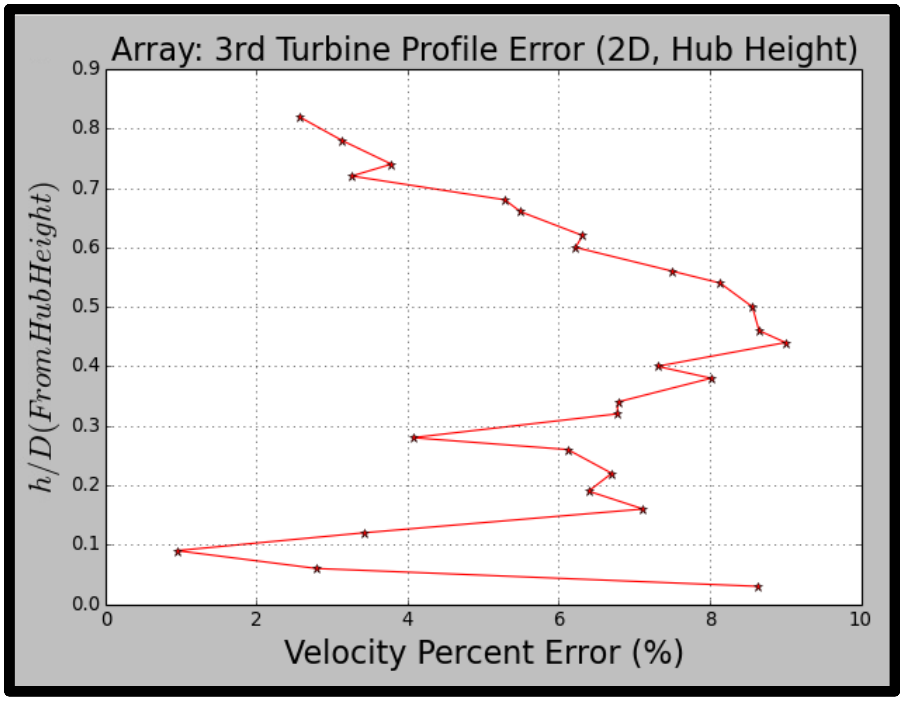


Figure 25: Percent difference between simulated and experimental velocity profile at 2D

The percent difference for the velocity profile at 2D downstream from the turbine and actuator disk was determined to be under 10 percent throughout the overall profile. The percent difference at 4D downstream can be seen in *Figure* (26).

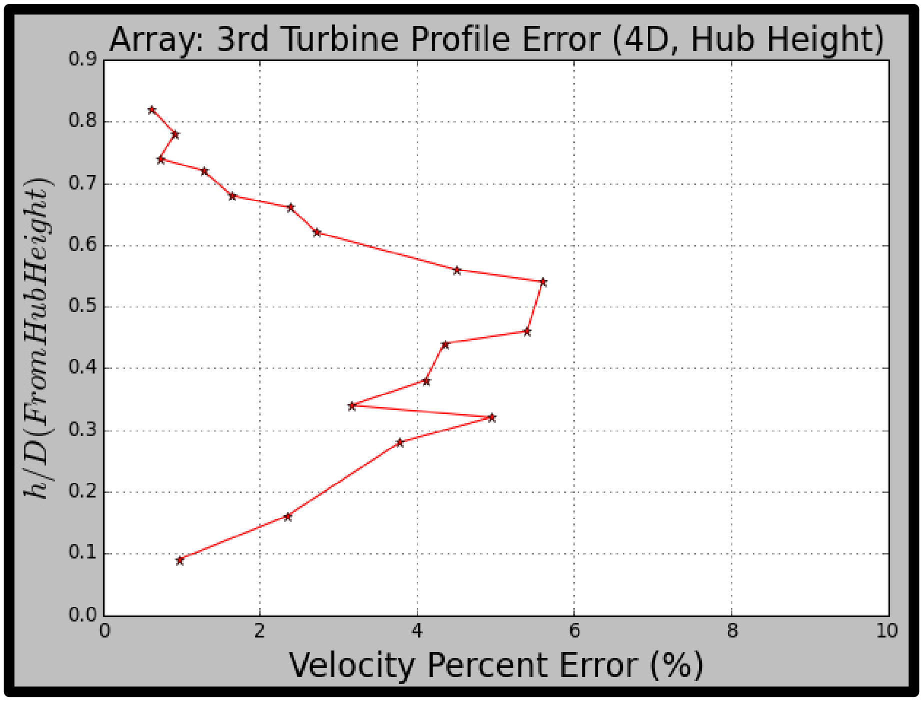


Figure 26: Percent difference between simulated and experimental velocity profile at 4D

The percent difference between the simulation and experimental data behind the third actuator disk and turbine was found to be less than 6 percent for the overall profile at 4D downstream. This percent difference is expected to decrease as the distance downstream the wake increases since the accuracy of the individual actuator disks were found to match the model wind turbines and porous disks better with a greater distance downstream.

Conclusion

The first stage of an independent numerical model was created where the model in its current state is able to match far wake trends of experiments; utilizing the specific experimental boundary conditions and model wind turbine C_p and C_t inputs. The data resulting from this model trends well if compared above hub height of a porous disk or model wind turbine, but below this there are at times major differences. This model is currently not independent of experimental boundary conditions and C_p and C_t inputs that have to be obtained experimentally and thus can only be compared to already done experiments. The major inaccuracies of the model occur in the flow closer to the wall. This is due to the use of the slip condition due to the failure to resolve that area using the k-ε turbulence model while using a no slip boundary condition.

For this model to become independent of pre-obtained experimental data, relationships between the characteristics of the results need to be recognized. For example, a better understanding of the relationship between the amount of turbulence injection and the incoming flow or the relationship between the C_p and C_t inputs to the array spacing. Once these relationships are defined, they can be applied to the model and will provide an accurate representation of any spacing of the turbines as long as they are positioned directly behind one another. This will allow researchers in the FPF to get information about array configuration much more quickly and will allow them to narrow down the experiments they will actually need to perform. This unfortunately would not work if the actuator disks were to be misaligned from the leading one.

In order to build an independent, numerical model that can be used to do optimization array studies, a different approach would need to be followed. An actuator line model instead of an actuator disk should be developed. This is because the actuator line better represents the effects of a turbine on the flow without the requirement of C_p and C_t input data and turbulence injection to match an actual wind turbine. The lines in this model generate the turbulence that is caused by the blades of an actual turbine, so the turbulence kinetic energy injection for the actuator disk model would be unnecessary. The use of the k-ω SST turbulence model would fix problems in the near wall regions since it wouldn’t require as much refinement and the use of wall functions and thus a no slip boundary condition, which represents better the actual problem, would be used.

By improving the computational power of the CFD cluster, it would be possible to make more accurate models by increasing the refinement of the mesh or maximum domain size. This would allow the application of the model to larger arrays but also help with resolving regions in need of high mesh refinement. The model could also be put into an algorithm to optimize the array spacing and orientation for maximize power output, and minimum load on the turbines.

References

[1] Barthelmie RJ; Pryor SC; Frandsen ST; Hansen KS; Schepers JG; Rados K; Schlez W; Neubert; Jensen LE; Neckelmann S (2010) Quantifying the Impact of Wind Turbine Wakes on Power Output at Offshore Wind Farms, J. Atmospheric and Oceanic Technology 27:1302-1317

[2] Beland, Kyle, Jeremy Bibeau, Christopher Gagnon, and Jacob Landry. "Offshore Wind Turbine Array." (2013). Print.

[3] Blazek, J. (2001). Computational fluid dynamics principles and applications. Baden-Daettwil: Elsevier.

[4] Shaw, C. T. *Using Computational Fluid Dynamics*. Prentice Hall, 1992

[5] Turner, John. (2014, August 6). Physical Model Study of the Wind Turbine Array Boundary Layer. FEDSM2014.

[6] United States Department of Energy: 20% Wind Energy by 2030: • Increasing Wind Energy's Contribution to the U.S. Electricity Supply DOE/GO-102008-2567, July 2008, http://www.nrel.gov/docs/fy08osti/41869.pdfhttp:

[7] Versteeg, H K, and W Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Harlow, England: Pearson Education Ltd, 2007.

[8] Wind Resources and Transmission Lines. Digital image. National Renewable Energy Laboratory. 19 Sept. 2014. Web.

Appendices

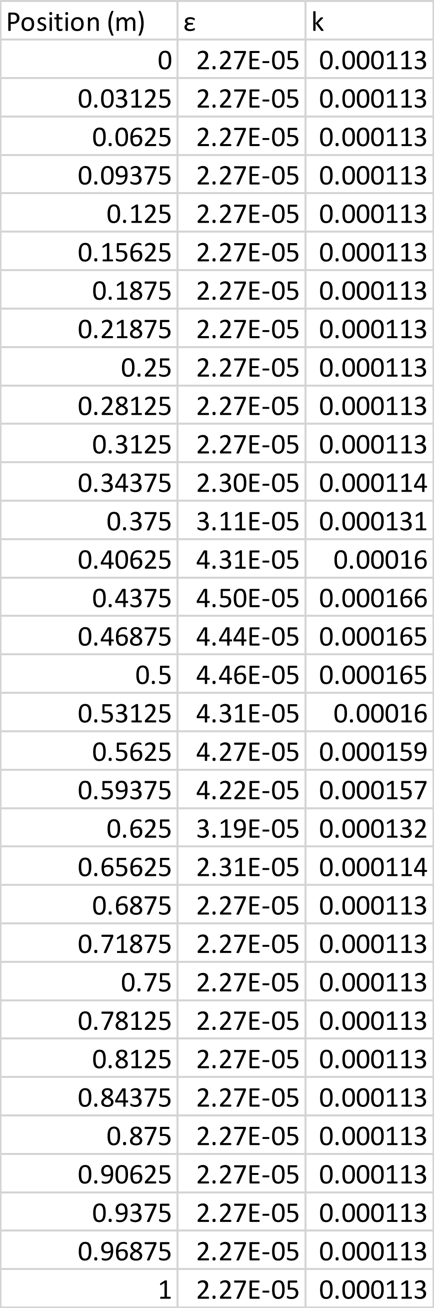
Numerical Model Code

The code utilized for our models can be found on our *GitHub* repository. Here you will find code for freestream and boundary layer flow with a single actuator disk and then code with a boundary layer flow with a 3x1 array of actuator disks. Navigate the different branches for the different codes.

https://github.com/tomkroll/RANS_OffshoreWindTurbines.git

Domain Inlet Data

This data was used to determine the inlet boundary condition to use for the freestream simulations. It was created from the numerical solution of the uniform wind tunnel flow to this point of 34.5 m in the wind tunnel. This data is in the z-direction and is constant in the y-direction.



Position (m)	ε	k
0	2.27E-05	0.000113
0.03125	2.27E-05	0.000113
0.0625	2.27E-05	0.000113
0.09375	2.27E-05	0.000113
0.125	2.27E-05	0.000113
0.15625	2.27E-05	0.000113
0.1875	2.27E-05	0.000113
0.21875	2.27E-05	0.000113
0.25	2.27E-05	0.000113
0.28125	2.27E-05	0.000113
0.3125	2.27E-05	0.000113
0.34375	2.30E-05	0.000114
0.375	3.11E-05	0.000131
0.40625	4.31E-05	0.00016
0.4375	4.50E-05	0.000166
0.46875	4.44E-05	0.000165
0.5	4.46E-05	0.000165
0.53125	4.31E-05	0.00016
0.5625	4.27E-05	0.000159
0.59375	4.22E-05	0.000157
0.625	3.19E-05	0.000132
0.65625	2.31E-05	0.000114
0.6875	2.27E-05	0.000113
0.71875	2.27E-05	0.000113
0.75	2.27E-05	0.000113
0.78125	2.27E-05	0.000113
0.8125	2.27E-05	0.000113
0.84375	2.27E-05	0.000113
0.875	2.27E-05	0.000113
0.90625	2.27E-05	0.000113
0.9375	2.27E-05	0.000113
0.96875	2.27E-05	0.000113
1	2.27E-05	0.000113