Senior Design Project
Wind Turbine Quantitative Analysis and Blade Geometry Derivation

Salvatore Ciminello (stc92)

MAE: 4021

Executive Summary

What are the desired functions of your design?

The desired function of the design is to make the wind turbine as efficient as possible. With regards to this we consider the power coefficient and output of the turbine in the design to be paramount. Along with that we optimize the design for high power output at a certain tip to speed ratio λ_r. Overall the desired function of the design is to help increase the power output for wind turbines so they are both more economically beneficial for use.

So overall, the general function of the design is to convert the incoming wind energy into mechanical shaft power as efficiently as possible, and this is through optimizing the chord length and angle distribution along the span of the turbine.

What constraints related to the main functions must your design satisfy?

Our design must satisfy several constraints, with regards to the physical design, operation, and geometry.

With regards to physical design, we need to consider that the blade, realistically, needs to withstand the normal loading suspected in a wind farm. When considering this we looked at the flap and edge wise bending moments, to ensure that the blade is thick enough to not succumb to ultimate failure. This was very important in the design conditions because if the turbine isn't fit to work in real conditions then having an optimal power output wouldn't really be helpful.

In terms of operations we were also constrained by having a practical limit on the tip speed ratio, such that it is constrained to a sensible range of λ_r. This is mainly because in the physical world if the tip speed ratio is too small, then rotational friction from the bearing overcomes the torque generated, but also at high tip speed ratios the turbine can either hit resonance and spin itself out of control, or the turbine just wears itself down way too quickly.

Finally in terms of geometry, so we can compare our design with the course design, we were constrained to use the same airfoil cross section, along with a similar radius, chord length, and twist distribution. This is especially the case with the radius of the turbine blade, since most of the power of a turbine comes from increasing the length of the blade, for consistency we kept it nearly the same.

What are the performance objectives of your design?

Our specific performance objectives were: Maximizing power coefficient C_P for a given tip speed ratio, select an optimal operating tip speed ratio, and to increase the overall mechanical power output from our design. All of those also depended upon the more abstract objective that we want to create both simulations, and calculations that reflect real world performance for wind turbines, without using an extremely convoluted method.

What alternative design concepts were considered?

Several alternative design concepts were initially considered. These alternative designs built off the governing principles of wind turbine physics. The first thing that was considered was changing the number of blades to be different from the standard 3 blade design. The rationale behind increasing the number of blades is because it will reduce the overall flapwise stress on a singular blade, by basically distributing the force due to the wind across more blades. The reason we did not decide to go along with this was because in industry, there is a notion that it is best cost-wise and in terms of infrastructure for the number of blades to stay at the standard 3. Along with that there is somewhat of diminishing returns when you increase the number of blades, with respect to optimizing the power coefficient, and minimizing the overall weight of the collective turbines.

Another design consideration was changing the airfoil type, either replacing it completely with another type of airfoil, or maybe changing the profile along the span of the turbine. This could help in optimization, giving us another parameter to control, which could've helped us find a better ratio between lift and drag coefficients to further optimize power, but ultimately we decided realistically, manufacturing wise it is a better idea large scale to keep the same and constant airfoil for the entire span of the length of the turbine.

What analyses were used to select among these alternative design concepts?

The alternative design concepts were put under the same analyses as the rest of the turbine, which was under the blade element moment theory model, and depending on the results, we assessed how influential changing these parameters will influence the rest of our design, while still paying close attention to adding to the complexity of the model. Overally, this really just meant doing the same analysis of the turbine, getting the results, and then apply the assumption that if we were to make our design for a real wind farm, will it still be advantageous for us in the long run. Overall the results of this analysis yielded that although increasing the number of blades could increase the overall efficiency of the turbine, and changing the airfoil at a certain annual section of the turbine could also improve power output, both will either increase the complexity too much for responsible change to the model, or will degrade other design objects not specifically evaluated here. An example of this would increase the radius and number of turbines although would greatly increase power output, it will make any wind turbine heavier and more susceptible to buckling at the base or increase the blade erosion at the tip.

What industry or society standards were used to inform or evaluate your design?

As stated above, the industry standard of using a constant profile for the airfoil, and holding the standard 3 blades for the design was used to both inform and evaluate our design. Although analysis was done on both of these topics, we decided to agree with the industry standards because they simply are a standard for a reason.

Another standard that was also somewhat taken from industry is the fact that we designed a conventional horizontal wind turbine that works on the principle of converting 1D fluid flow into energy, and yaws to be perpendicular to the direction of the wind speed, instead of developing a vertical wind turbine, that works for unidirectional flow. Simple we made this decision for the design because firstly, the course is designed to analyze horizontal turbines and not vertical turbines, and secondly because vertical turbines, although convert unidirectional flow to energy, are a lot less efficient due to their large range of application. So similarly we agreed with the industry standard of designing a horizontal wind turbine, which is a lot easier to analyze from a fluid dynamics perspective. Below **Figure 1** shows the differences between a traditional horizontal wind turbine, with its vertical counterpart.

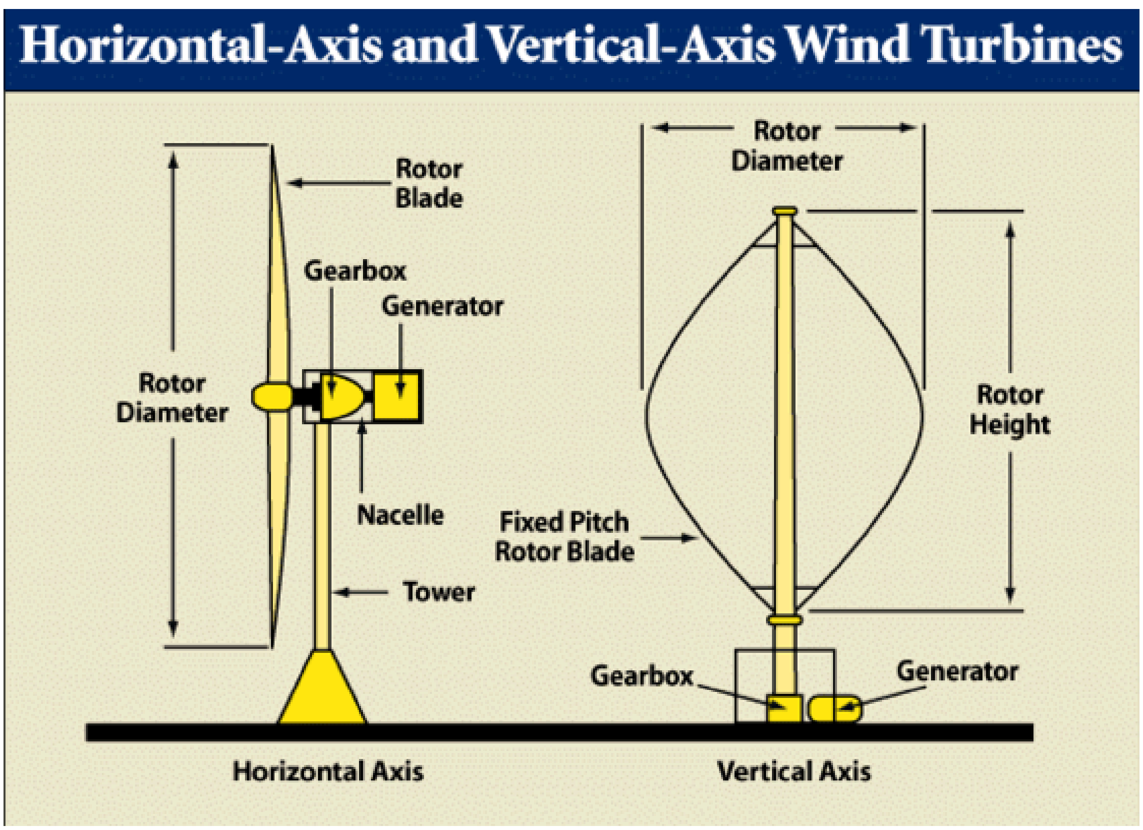


Figure 1: Difference between Horizontal and Vertical Turbine Geometries

Which concepts or skills learned in your coursework were applied to the design? Projects are expected to make substantial use of MAE and related ENGRD classes. Please provide a list with each entry providing the department and number of the course, plus a brief description of the particular concept or skill used.

A lot of skills were used from my undergraduate work to make and develop this design to the extent needed for this project. Below is a list of specific courses and the subsequent contribution they had to this project.

- MAE 3230 Introduction to Fluid Mechanics - This course gave me the resources to start understanding fluid flow, and toward the end of the semester gave me an introduction into pressure differentials from an airfoil in a uniform steady flow.
- MAE 4272 Fluids and Heat Transfer Lab - A large portion of this class was continuing with our study of fluid flow, especially in the turbulent regime which helped for this project. Along with that this class also made us look at turbines as a collection of different engineering fields, like statics, vibration analysis, along with the fluids component. I assumed a holistic approach to engineering throughout this project, not just understanding how to harvest the kinetic energy in the wind for power, but also understanding failure cases, and how that energy is converted to mechanical work.
- MAE 3270 Mechanics of Materials - This course built upon the introduction to statics class MAE 2020, and helped me further understand stress and failure modes of solid bodies. This helped for this project because it helped me understand the idea of failure in the turbine as it applies to fatigue, and how a good well rounded design for a capable wind turbine also helps resist cyclical loading. This course also gave me a brief introduction into FEA through Ansys: Static Structural, and Ansys Granta, which helped me build familiarity with Ansys: Fluent for this project.
- MAE 2030 Dynamics - This course built on the Newtonian physics learned in introduction to physics in freshman year. It greatly helped in understanding and developing my own free body diagrams, which skill I used in this class to understand the relationship between the forces on the airfoil, and how that can be translated to the useful torque for the turbine, and the normal bending stress on the turbine blade.
- Project Team - Although not a traditional course my project team first introduced me to CADing, and Ansys Static. Knowledge of both of these was very important in finishing this project, and more specifically manipulating the geometry of the given turbine for the fluid dynamics analysis.

Evaluate your design, relative to its function(s) and constraints. How well did your design meet each of the performance objectives? How well does your design compare to other, existing solutions to the problem?

Our design did a very good job in meeting our functions while conforming to the constraints. Our main objective was to create a new design for a wind blade for a turbine, that is able to efficiently and effectively convert kinetic energy in the wind into mechanical work that can be converted into electrical energy. We were able to complete this objective, while also exceeding in our goal by designing a wind turbine that more effectively converts wind energy in mechanical work than the standard blade design supplied to us.

I would say our blade design definitely fairs very well even compared to existing solutions. The one comparison we can empirically make is that our solution for the blade is objectively better than the existing blade we were given to compare against. I am sure there are other blade designs out there that are better suited for offshore turbines, or are better at preventing blade erosion, or are better in terms of infrastructure needed, or designing for stress loads, but for the purpose of this report, which was to optimize a blade for objective power performance, how design is very good. If we were to increase the fidelity of our design but also considering loading, or specific material specifications that could change the view, but as of now are design is better than many designs.

What impact do you see your design, if implemented, having upon public health, safety, and human welfare, as well as upon current global, cultural, social, environmental, and economic concerns?

If my design is implemented I believe there will be positive effects across the board with regards to public health and human welfare, along with global and environmental concerns. This is mainly due to the fact renewable energy sources like wind power, are being used and developed in an effort to help stop and/or reverse the effects of climate change due to overuse of non-renewable energy sources. By using these resources like coal, over the past few decades the average temperature of the earth has gradually increased. This increase in temperature leads to the accelerated rate of melting of the glaciers, rising sea levels, and more frequently extreme weather events.

The result of this climate change, not just makes current lives harder to live, but future lives nearly impossible with the trajectory we are heading. This rationale is exactly why the implementation of my design would both increase the overall health of the public, but also improve the overall human welfare by decreasing the rate of climate change affecting lives across the globe.

Economically speaking, creating more wind turbines does put a bit of a strain more on the market, because a lot of the resources behind wind turbine manufacturing is hard to come by, but overall once these wind turbines manufactured and installed, on average wind power is 7 times cheaper than other non-renewable sources, so in the long run it will be better for everyone all around.

What format did your design take? For example, is it a complete set of CAD drawings, a working prototype, a full finished product, a system configuration, a process map, or something else?

The format of my design as of now is developing the blade geometry, where I have a matrix that describes the geometry of the turbine along the span of the blade, with respect to the radius at that point, the chord length at that radius, and the local pitch angle as well. Although developing this model was complex, the overall finished design and geometry of the blade is rather simple. Further analysis and CADing can relatively easily be done, by just importing the airfoil 2D sketch into a fusion 360 file, and scaling the chord length for a given radius. Then orientate that annual section of the blade to the correct pitch angle, then loft the 2D planes together to create the entire span of the turbine.

Describe each student's role in the design project if it was a group project.

Not applicable. Although in this report the plural “we” is used multiple times, all of this work was done individually.

Project Overview

In this paper, from start to end, I will discuss and show the process of developing a wind turbine. To start I will walk through the physics that govern fluid flow around objects. This will help us understand pressure differentials, and the airfoil theory, whereas fluid flow around an airfoil encourages lift and drag. This physics then consequently determines turbine design. Using this model, I will then analyze a given turbine as it applies to computation fluid dynamics, along with experimental data taken from the blade.

After analyzing the blade, using the theory behind turbine development, I will construct and design a new blade that will also be analyzed rigorously through Ansys CFD and theory.

Lastly I will compare the two designs, and using theory, simulation, and practicality, I will determine which design is better.

Understanding Blade Element Momentum (BEM) Theory

To start our critical analysis of wind turbines it is first necessary to understand the theory underpinning all of turbine analysis which is the blade element momentum theory (BEM). This theory builds off the basic idea of airfoils in basic fluid dynamics, whereas we can analyze the pressure differential of a 2D foil, and how air or any fluid moves around the foil, assuming one direction steady state flow. BEM builds off this theory by integrating across the 3D length of the turbine, by summing up the individual contributions to the lift and drag forces due to the 2D airfoil slices. By enforcing these aerodynamic forces on the individual 2D cross sections, we can understand and determine the thrust, torque, and overall power of the turbine.

Aerodynamic Forces in BEM

To start with BEM, we first look at the aerodynamic cross section of the airfoil that makes up the turbine, and we define geometry that will be helpful in determining relative forces. Below is a figure that helps define relevant geometries for the turbine that helps us determine incremental forces and torques.

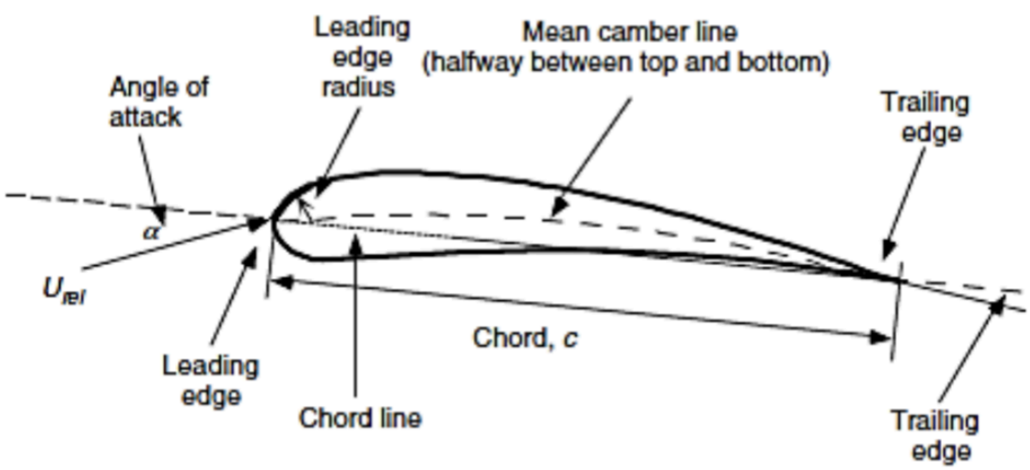


Figure 2: Geometry for Airfoils

This figure helps us define geometries and planes that will be useful in understanding the forces in this equation. Following that the next assumption that is made is that if we have a wind velocity coming from U_{ref} there will be two aerodynamic forces at play. These forces include the lift force that will be perpendicular to the wind speed U_{ref} and a drag force that will be in the direction parallel to the wind speed U_{ref}.

Overall this derivation comes from bernoulli principle, whereas if we have a fluid flow approaching an airfoil, since the system can be assumed to be isentropic and steady state we know the bernoulli constant will also stay the same, such that:

$$p_1 + \frac{1}{2}\rho {U_1}^2 + \rho g z_1 = p_2 + \frac{1}{2}\rho {U_2}^2 + \rho g z_2$$

$$p + \frac{1}{2}\rho U^2 = constant$$

Applying this to the airfoil modelled, it will mean that in places of high fluid speed, the pressure will be low, and at places of low fluid speed, the pressure will be high. Using the idea of conservation of linear momentum we can then surmise that if we have an airfoil with a pressure differential, the airfoil will have a lift force in the direction of that decreasing differential, and a drag force perpendicular to the differential.

Finally if we were to finish up the model by applying the aerodynamic force equations, along with considering the direction of the application we will learn that for a given aerofoil the following equations on motion hold true.

$$dF_l = C_l \frac{1}{2}\rho {U_{rel}}^2 cdr$$
$$dF_d = C_d \frac{1}{2}\rho {U_{rel}}^2 cdr$$

To then finally finish up this model we need to remember we are modeling the entire turbine and not just the airfoil cross section, so we must reference angles that relate the relative wind to the current position of the airfoil, along the span of the turbine. We can also then break these two forces into components that will tell us the normal and thrust force, which later will be explained to give us the shear bending moment, and the useful torque for power generation in the turbine.

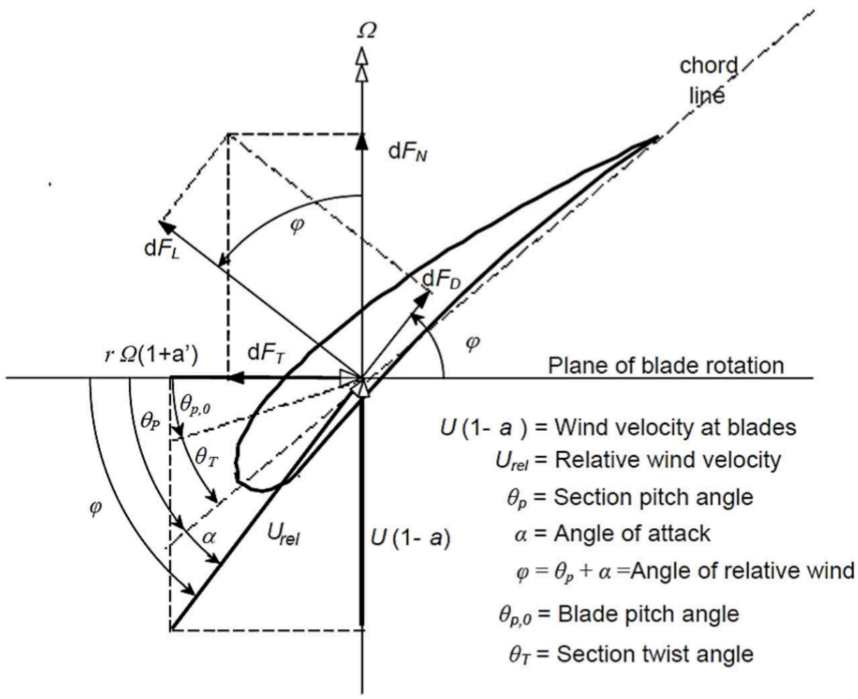


Figure 3: Airfoil Turbine Angle Geometry

Referencing the above figure, **Figure 3: Airfoil Turbine Angle Geometry,** we relate the current orientation of the airfoil along the span with the constant free wind speed. It is also important to note that the reference wind speed takes into account the rotation speed of the turbine at that point, and is the magnitude of the free-wind speed and the rotational speed of the turbine.

Superimposing the incremental drag and lift forces dF_d and dF_l then gives us dF_N and dF_T, which is the vector sums of the drag and lift forces in a useful coordinate system. dF_N being the incremental contribution of the airfoil in the direction normal/parallel to the fluid flow, bending the turbine in the direction of the flow, while dF_T is the incremental contribution of the airfoil on the thrust of the airfoil, in the direction that will do rotational work with respect to the hub of the turbine as torque.

Finalized BEM Equations / Wrapping it up

Finalizing this model using the above relations, we can develop an equation that tells us about all the incremental additions to the normal force and torque in the form of an differential equation whereas;

$$dF_N = \frac{1}{2} B\rho U_{rel}^{\ 2}(C_l cos(\varphi) + C_d sin(\varphi))cdr$$

$$dQ = \frac{1}{2} B\rho U_{rel}^{2}(C_{l}sin(\varphi) + C_{d}cos(\varphi))crdr$$

Next for this method we apply a few assumptions. We assume the power coefficient is in the optimal regime, there is no wake rotation, and then we also pick our design conditions for the radius of the turbine, number of blades, and tip speed ratio. Then using the above equations we finally come up with the governing equations for blade design.

$$U_{rel} = \frac{2U}{3sin(\varphi)}$$

$$\frac{BC_{l}c}{4\pi} = sin(\varphi)tan(\varphi)$$

$$tan(\varphi) = \frac{1-a}{(1+a')\lambda_{r}} = \frac{2}{3\lambda_{r}}$$

After deciding on the above, along with choosing an airfoil profile that will be held constant throughout the span of the turbine, we can then use computer simulations and calculations to directly give us the needed parameters for design such as: pitch, angle of attack, chord length, and radius.

This finalizes our explanation of blade element momentum theory, which we can use to find an optimal design for the wind turbine, and will be used later on in this paper when developing our own wind turbine.

Analysis of Course Turbine

Introduction

The course turbine is a three-bladed, horizontal axis wind turbine. This turbine has a blade radius of 0.305 m, with a variable blade pitch ranging from 18.9 degrees to -2 degrees. The cross section is also made up of a constant airfoil NACA 2414, and also has a variable chord length that decreases as the length increases.

Detailed structural and material dimensions were not provided for the model so a full finite element analysis of the blade was not performed. For the remainder of the project we will work under the assumption that the turbine will not deform significantly or reach a failure point under the suspected loading.

Computational Fluid Dynamics Model (CFD)

For the computation fluid dynamics approach, we developed a Ansys Fluent simulation. This simulation works on the principle of importing blade geometry as a surface, and building a volume around that surface where the fluid is simulated to be. This fluid volume was large enough to make the assumption that at the boundary conditions of the volume the fluid speed is constant, which follows along with our assumption that far into the wake of a turbine the fluid speed returns to normal. Below is a figure showing the defined geometry for the analysis

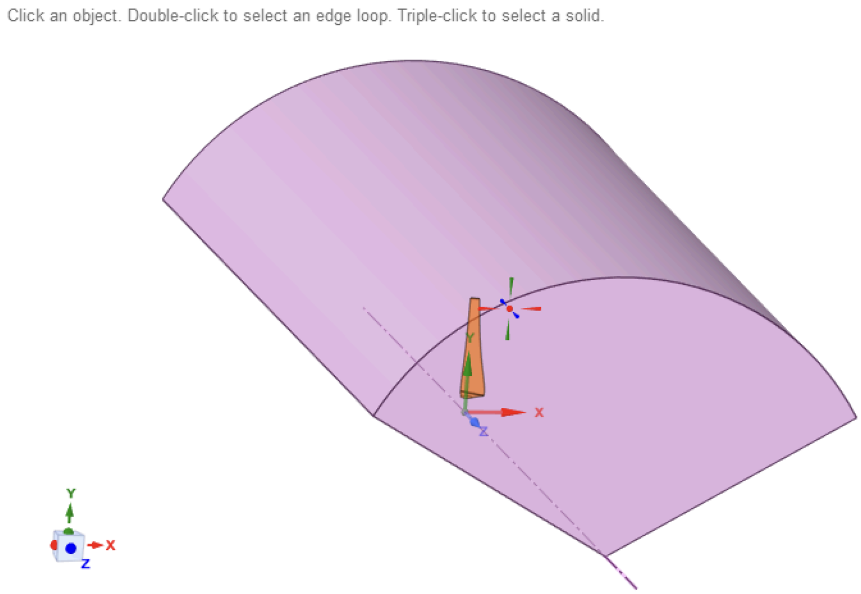


Figure 4: Blade Geometry and Fluid Volume

The exact process of setting up the actual physical setup of the simulation is as follows. First after making the geometry and making the correct meshing for the geometry you first specify the model you want to use when doing the calculations. For this analysis as the tutorial showed we used the GEKO model. After this we then checked the boundary conditions that ANSYS Fluent automatically assigned to the fluid volume and body. With regards to this ansys automatically assigned the blade to be the wall face, the conical side to the an inlet, and other sides to be outlets inlets and periodic boundary conditions. Finally to define and be able to change a TSR we picked the fluid to have a rotating reference frame at a speed along the z-direction, flow direction. Another more advanced technique at this step that I used a few times was to turn this value to be a parameter, and do a parameterized sweep, where you changed the TSR so it iterates over every value in a vector you define. The final step is then defining the wind speed: the inlet, the inlet top, and the initialization before calculation. Then you can finally solve it. Each solution should take about 10-20 minutes depending on the meshing. In a parameterizing sweep every iteration of the parameter normally has the simulation run faster because we know where convergence is likely to occur.

To analyze the cases of different pitch angles, we would go into the original geometry file, and rotate the blade slightly, either rotating it a few degrees in the positive or negative direction with respect to the wind in the z-direction.

A mesh was then constructed, and this mesh depending on how rigorous our specific analysis was either coarse or fine in its element sizing. Finally after mesh construction was applied we set up the experiment environment in ansys. This included defining the relevant boundary faces, such that the inlet, outlet, and other faces of the enclosed fluid volume had the appreciable physical conditions. Lastly the blade surface was modeled as a solid wall region, whereas the fluid around it was modeled as a rotating region, with a moving reference frame, corresponding to the correct tip speed ratio TSR. We then ran this simulation until we hit convergence of the model. Below is a figure for one of the examples that shows the number of iterations, along with the precision of the converging model.

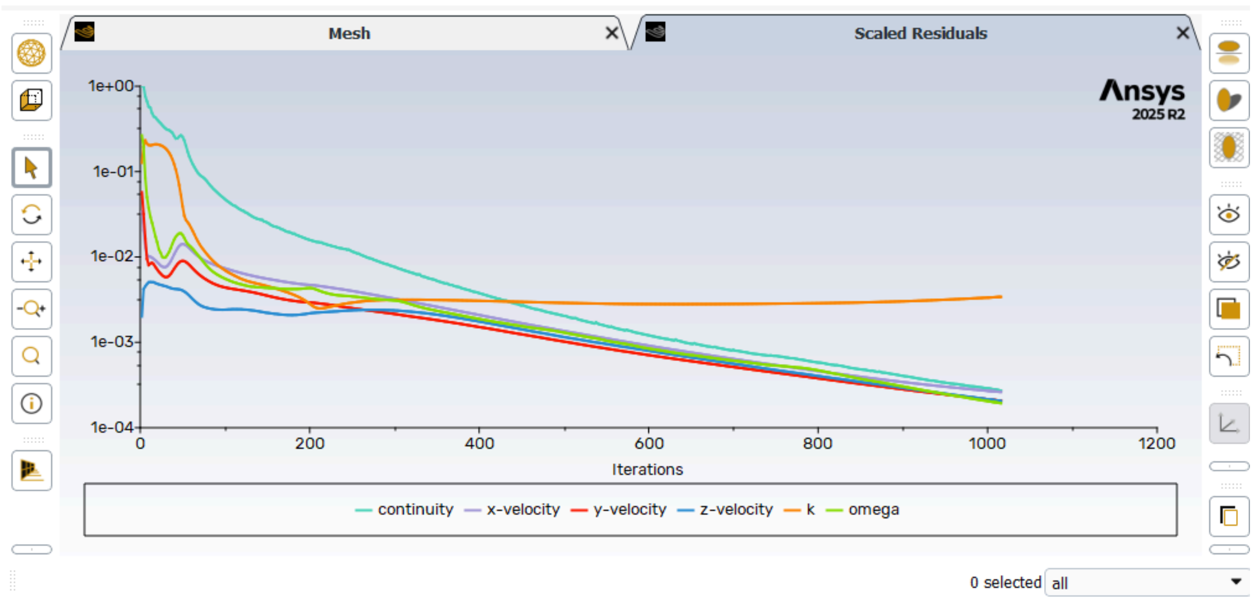


Figure 5: Convergence of Turbine Model in CFD

Figure 5, Convergence of Turbine Model in CFD, displays the residual history of the turbine in the CFD simulation. The residuals represent the fluctuations and imbalances in each of the governing equations of the turbulent flow during all the hundreds of iterations of the solving. In our case the continuity and momentum equations decrease by a good margin over the set of iterations, which indicates that we reached a good level of convergence in our field. Overall the behavior shown in this graph demonstrates a stable converged solution.

For the remainder of the CFD tests a few things were changed. Firstly, depending on what pitch angle we were analyzing, we would have to go into the original geometry in spaceclaim and rotate the airfoil slightly about its pitch angle. Due to this each time we changed the pitch angle we would have to generate a new mesh.

Another thing we would have to change is when we were analyzing the different tip to speed ratios, we would have to manually go into the simulation after each test and input in another angular speed that mimics the changing tip to speed ratio.

Results from CFD

From the model, for a given pitch angle we would convert the TSR for the turbine into an relative angular speed of the air. This conversion was implemented using the following formula:

$$\omega = Angular\ Speed = \frac{\lambda_r \times U}{R}$$

Whereas, U is the free-stream velocity of the fluid field, λ_r is the tip to speed ratio, and R is the overall radius of the turbine. After getting this angular speed, as stated above, we applied it to the rotating frame for the fluid of the air in the ansys simulation, then we started the simulation. After then letting the simulation run, we went into the post processing part of the ansys simulation, and were able to get out the torque. We then related this torque per blade, to get the overall power of the rotor, which we then used to calculate the power coefficient.

$$P_{rotor} = \tau \times \omega \times B$$

$$C_P = \frac{P_{rotor}}{P_{air}} = \frac{\tau \times \omega \times B}{\frac{1}{2}\rho A U^3}$$

Using this method we were able to extract a power coefficient from the CFD model that we can use to compare with experimental results.

Below are two instances of comparison between the experimental data and the CFD results. The two instances include the standard blades original pitch angle of $\theta_p = 4^\circ$ at $U = 6m/s$ and also the standard blade at a modified pitch angle of $\theta_p = -2^\circ$ at $U = 6m/s$. Further analysis was done in Ansys Fluent that will be touched on later, but this was done to compare the experimental results with the theoretical results. Along with that this shows with coarse mesh the pressure distribution of the turbine in a plane $0.1m$ away from the hub, as you change the TSR.

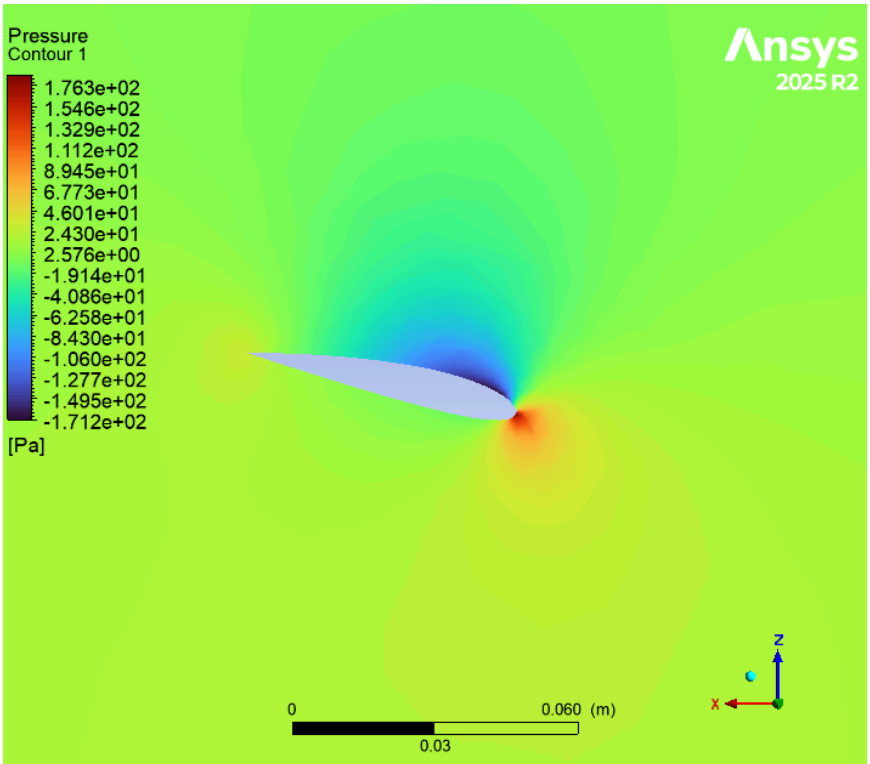
Pitch Angle 4 degrees, 6m/s

TSR λ_r	Angular Speed (rad/s)	Torque	Power Rotor	C_p
2.5	49.180328	0.023818	3.514057	0.090963
3	59.016393	0.028372	5.023186	0.130028
4	78.688525	0.039744	9.382119	0.242861
5	98.360656	0.040608	11.982600	0.310176
6	118.032787	0.032038	11.344674	0.293663
7	137.704918	0.016985	7.016837	0.181635
8	157.377049	0.000736	0.347353	0.008991

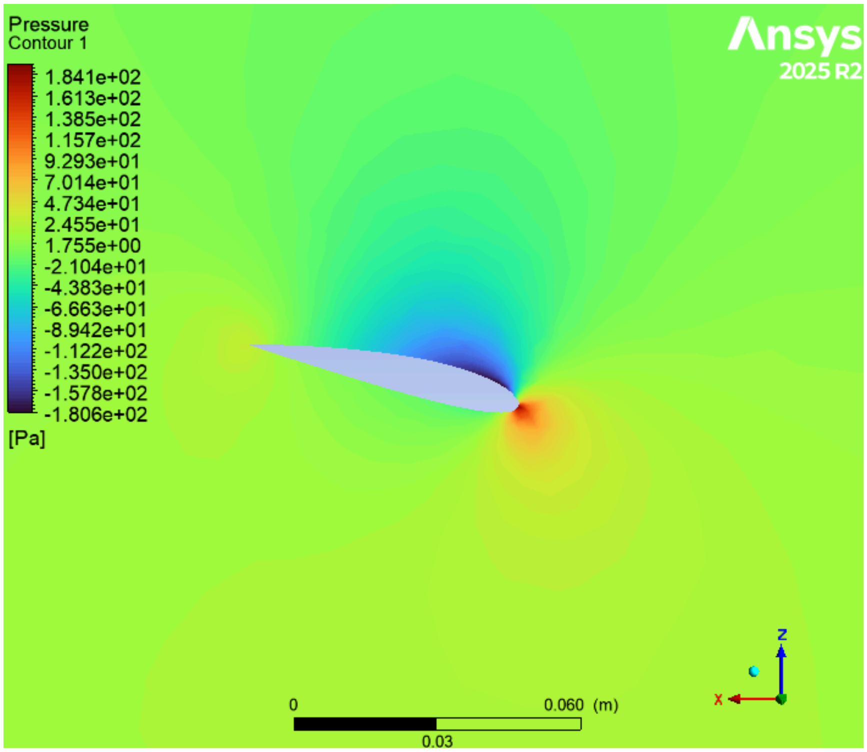
Pitch Angle -2 degrees, 6m/s

TSR λ_r	Angular Speed (rad/s)	Torque	Power Rotor	C_p
8.93	175.6721311	0.000541313	0.285280825	0.007384649483
8.63	169.7704918	0.00348019	1.772500704	0.0458821458
8.19	161.1147541	0.00829	4.006923934	0.1037214077
7.65	150.4918033	0.011996	5.415899016	0.1401934949
6.98	137.3114754	0.0147359	6.070224511	0.1571310666
6.29	123.7377049	0.0153342	5.692256144	0.1473471496
5.97	117.442623	0.0160431	5.652431233	0.1463162601
5.3	104.2622951	0.0160481	5.019635213	0.1299359906

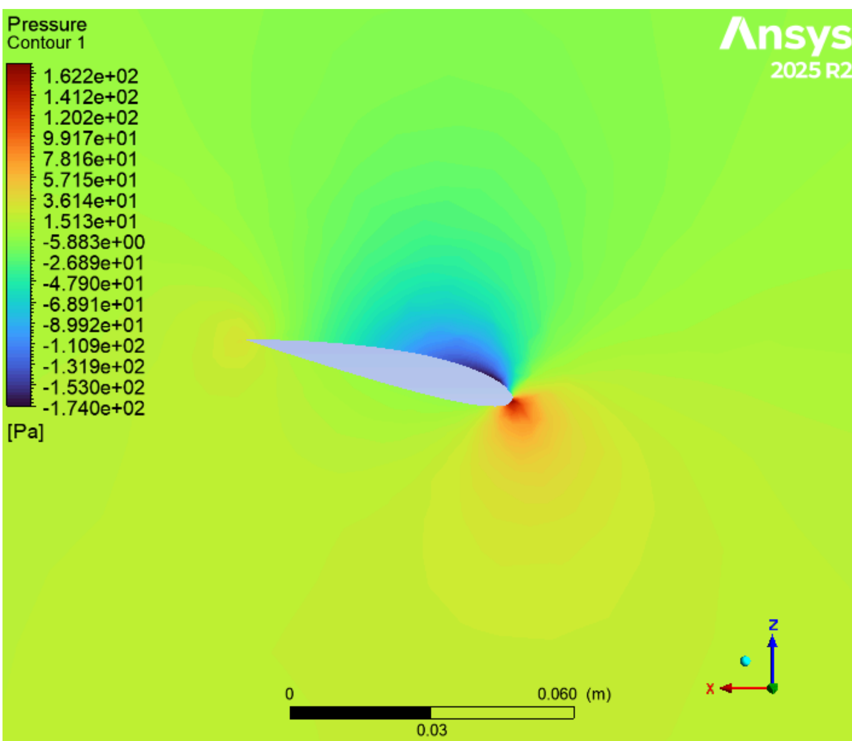
$\lambda_r = 8.93,\ \omega = 175.7\ rad/s$



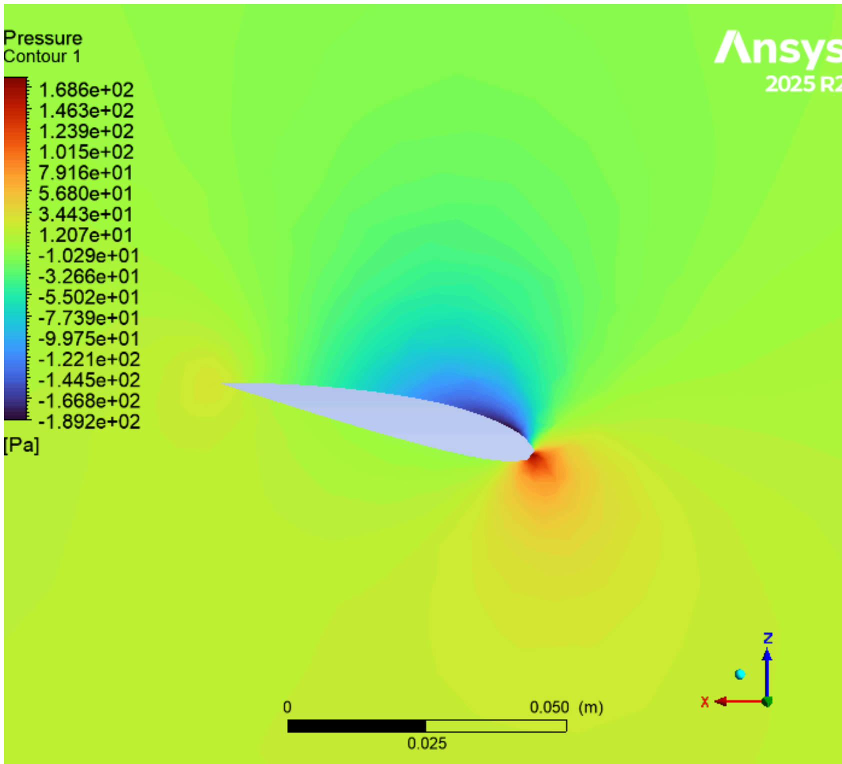
$\lambda_r = 8.63,\ \omega = 169.8\ rad/s$



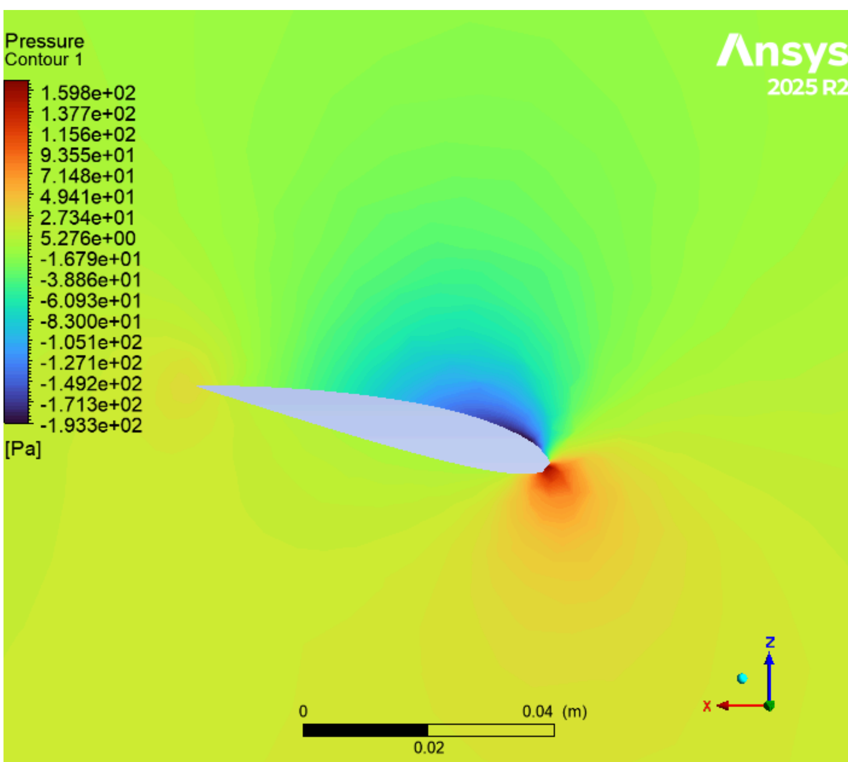
$\lambda_r = 8.19,\ \omega = 161.1\ rad/s$



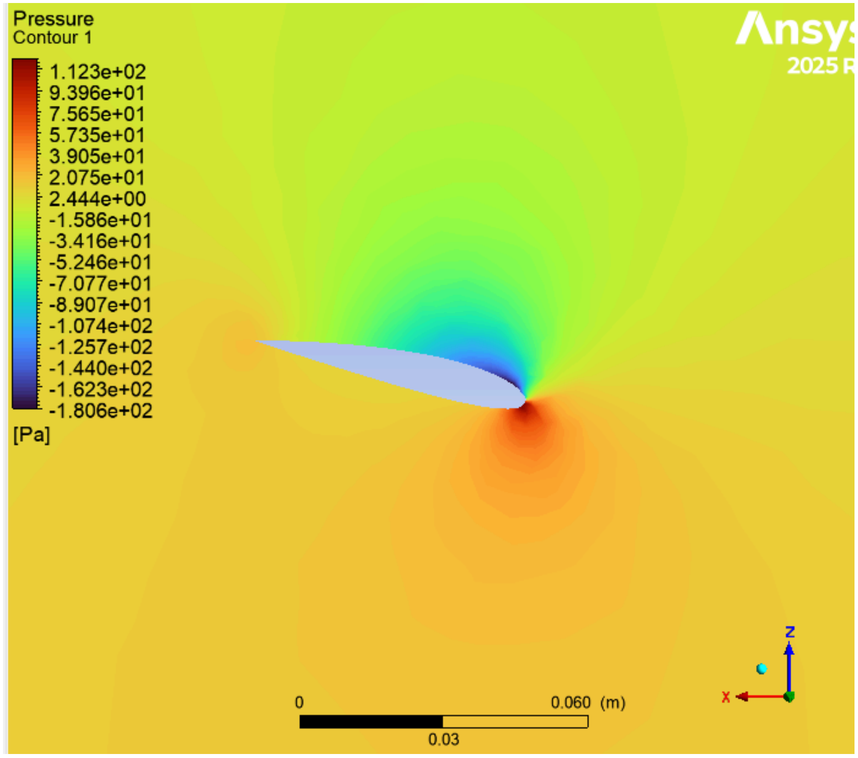
$\lambda_r = 8.19,\ \omega = 161.1\ rad/s$



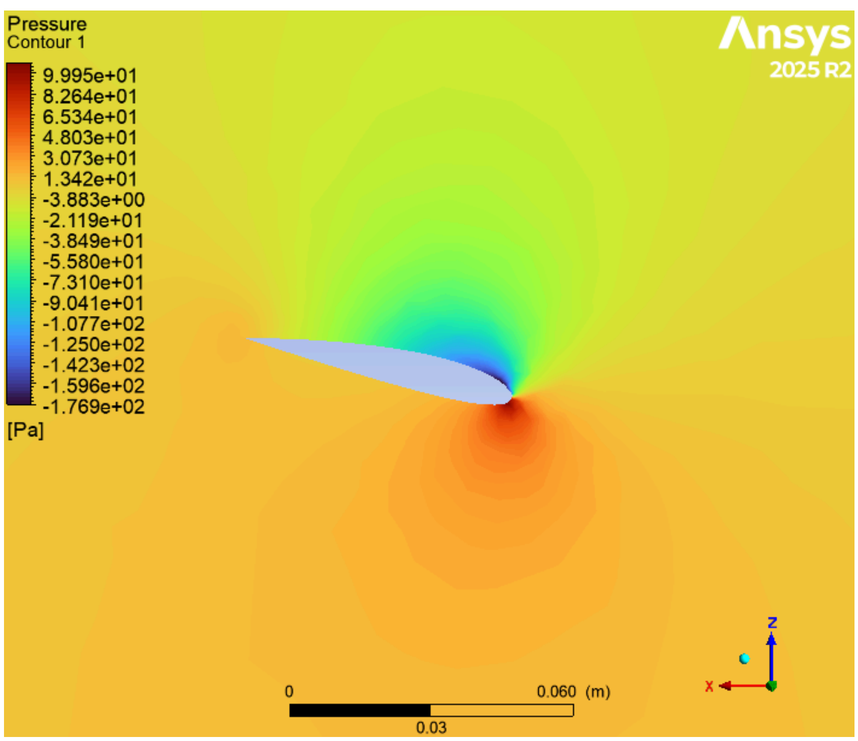
$\lambda_r = 7.65,\ \omega = 150.5\ rad/s$



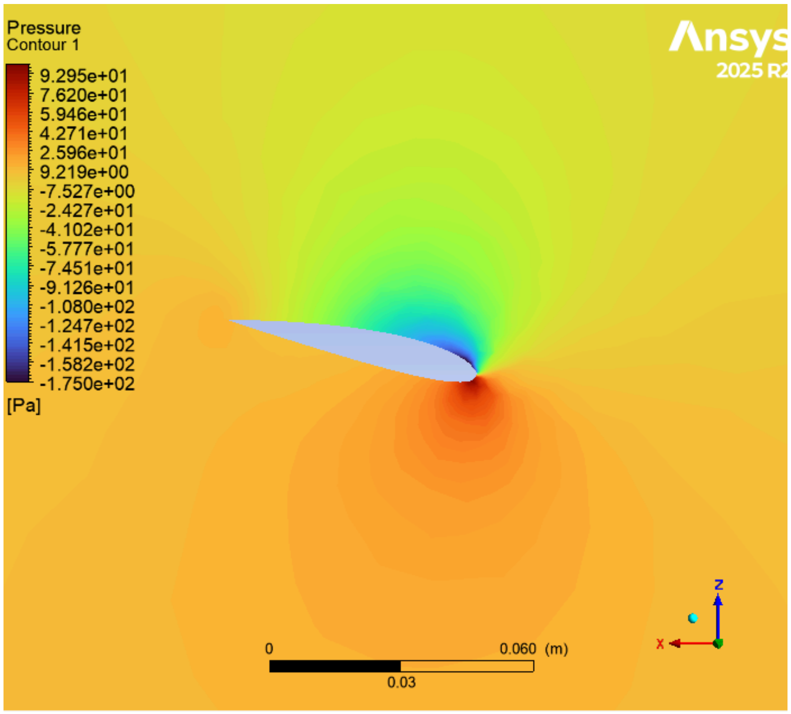
$\lambda_r = 6.98, \ \omega = 137.3\ rad/s$



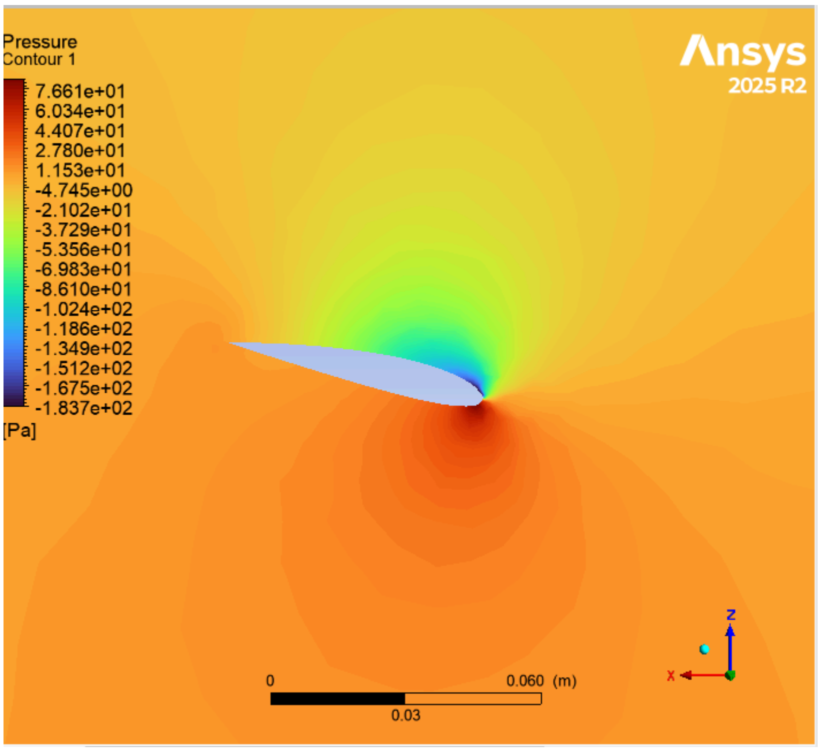
$\lambda_r = 6.29, \ \omega = 123.7\ rad/s$



$\lambda_r = 5.97,\ \omega = 117.4\ rad/s$



$\lambda_r = 5.3,\ \omega = 104.3\ rad/s$



Aerodynamic Testing of Blade

The experimental data from the aerodynamic testing of the blade in this report used for comparison was provided to us by the MAE 4021 measurements. This dataset contained power coefficients, for aerodynamic testing of the wind turbine overall geometry in a wind tunnel. This wind turbine has the same geometry for all testing sets, but the difference in the measurements come from the following: changing the pitch angle for the overall blade, changing the resistance on the torque brake, and changing the wind speed in the tunnel. From the dataset this quickly changed the power coefficient across the board depending on which factor changed. We hand picked which data sets to compare with our blade and with the CFD. We did not compare and test all data sets because this will take a lot of time, and we also only wanted to test and compare the datasets that gave us tangible differences and high power coefficients. This was a part of our decision because we already saw from the below CFD results, **Figure 6**, that normally with coarse mesh, the power coefficient will go down, especially out of the designed range of 0-4 degrees for the turbine pitch angle. So we took this into account but only comparing results that were empirically going to have a high power coefficient, that we expected would decrease, so we can overall analysis the degradation due to coarse mesh analysis, along with comparing the overall trend lines between the models.

Comparison Between CFD and Experimental Data

Following rigorous simulation in Ansys Fluent and the testing of different blade geometries, tip speed ratios, and free-stream wind speeds we came to the conclusion that although generally Ansys Fluent in CFD can simulate extremely complex environments really well, at some points the simulation fell off and gave us data results that don’t line up within a degree of responsible accuracy to the experimental data supplied.

Multiple tests were done in Ansys Fluent to test the validity of the results and see how well we can correlate the ansys fluent CFD relates to the testing data. In the following data set I tested the same blade geometry that was tested in the lab, set at an angle of -2 degrees and in a free-stream wind speed of 6m/s. Then we swept the ansys fluent simulation and tested every other TSR that was tested in the lab. Subsequently we then got a torque from the Ansys fluent results page. This torque, like above, was then converted to the power extracted by the rotor, and then we were able to find the power coefficient based on that. We then plotted that power coefficient from this analysis, along with the same angle and data points but with a finer volume mesh, with the experimental power coefficients. Below is the data.

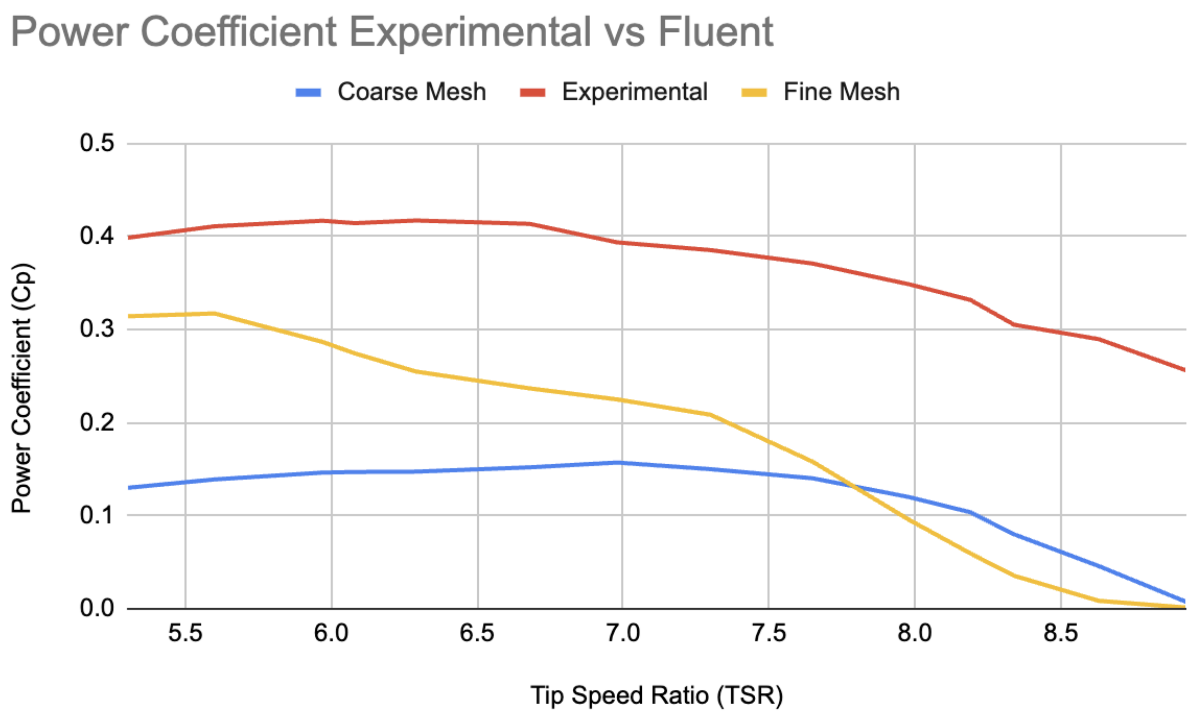


Figure 6: Experimental and Variable Mesh Power Coefficients Comparison

The data above shows how the overall simulation of a given fluid volume is not always accurate. When discussing this with Professor Bhaskaran he showed that having coarse mesh will not always produce the expected results that might come from the experimental set up. And how if you were to increase the mesh density by a bit this can either translate to being a lot closer to experimental results, or might not have any effect at all. Overall this graph and discussion goes to show that although CFD is a useful tool, it can misrepresent a turbine by a large factor if the meshing isn't extremely fine. For the sake of time, I tried to keep each wind speed and TSR to be a 10 minute long simulation at most, so maybe if I spent more time simulating then the results would've been a lot closer to the experimental results but this wasn't the case. As seen above though the overall trend is that as meshing density increases, the power coefficient and results of the simulation trend closer to the experimental values.

Designing a Turbine Using BEM

At this portion of the project, using what we have learned with regards to blade element momentum theory, we will use the theory to develop and generate a rotor blade design and hopefully improve on the power output of the standard blade design. As explained above in the second section of this report, BEM is the intersection between one-dimensional momentum analysis and the two dimensional lift and drag forces on an annual section of the blade, and then uses that to predict the overall torque on the blade due to wind at a speed.

To design the turbine we had to first consider the constraints for the geometry. The first constraint considered is the airfoil we are using for the profile of the design. Per the design requirements we used the NACA 2414 airfoil as that is the same airfoil that was used for the entire span of the experimental data wind turbine. The next constraint was that since we were comparing this design to the experimental one, we needed the length of the turbine, and the base radius to account for the hub of the turbine to be the same, so that meant overall we needed to keep the same overall minimum and maximum radius for the turbine as the standard.

Lastly for our design we decided to specifically analyze the performance of the blade at the wind speed of 6 m/s, with an air density of $\rho_a = 1.225\ kg/m^3$, with 3 blades.

Following that we then needed to implement all this into matlab for the further analysis, and to generate the overall geometry with regards to pitch and chord distribution. For this part we needed to discretize the radius, so we can evaluate the optimal blade pitch and chord length for each discrete point along the length of the turbine radius. We did this by building a vector that held at one end the minimum radius, the radius of the hub, and at the other end the maximum radius, then we made N equally spaced element between those two, to evaluate at. After that we then defined a certain deserved tip to speed ratio λ_{des} which we then used to calculate the local inflow angle for the fluid flow, which then relates us to the turbine pitch at that discrete radius. The following operation went like the following:

$$\lambda_r = \frac{\Omega R}{U_\infty}$$

$$\Phi(r) = \frac{2}{3} tan^{-1}(\frac{R}{\lambda_r r})$$

$$\theta_p(r) = \Phi(r) - \alpha$$

The whole process of this calculation is made in the Matlab script **comp_Cp_w_geometry** which is in the **Resources** section of this report. Using this script we were able to use a desired TSP λ_r, and using the optimal BEM model above, construct geometry that will be best suited for that TSP ratio case.

Following that another script, **optimal_design**, was made such that it takes the function **comp_Cp_w_geomrety**, and applies a range of different tip to speed ratios, and then gives us the best design for the optimal TSR, and graphs the power coefficient with respect to TSR. Using this method we were able to get a wind turbine geometry that gives us the highest possible power coefficient using the BEM model.

Modeling the Hub and Discretizing the Geometry

Through testing different models for the hub it became apparent that two factors played a key role in estimating blade's overall power coefficient. These two factors were, the size of the hub, and the number of discrete points along the blade. Both of these played a role in the blade BEM model for this key reason, sampling many points at the edge of the blade will make us have a larger power coefficient than if we were to sample points evenly along the length of the blade. This is because the majority of the power extracted from a wind turbine is harvested from the outer 30% of the span of the blade. This means if we were to: 1. Make the hub of the turbine (starting radius) of the blade really small, 2. Sample small number of points along the blade, our power coefficient will be smaller than the actual power coefficient. Simply, if we sample and look at only a few contributions along the blade, our expected blade power would be smaller than if we look at many contributions along the blade, focusing on the outer part of the blade.

With regards to this we compared the instance of starting the blade at at 5, 10, and 20% of the blade radius, and sample at 10, 20, and 40 points along the radius. We determined that the most accurate representation of both the true blade geometry, and losses due to a mechanical system can be seen best in the case of 10% radius start of the blade, and a sample of 40 points along the blade radius. This gave us the below power coefficient for the blade design.

BEM Design Results

Using the above method, the optimal design/geometry for the wind turbine calculated through matlab is below. This geometry is designed for and has the highest power output for a TSR $\lambda_r = 4.76$.

Geometry for best design:

Radius_m	Chord_m	Pitch_deg
0.061	0.066483	22.189
0.088111	0.05843	15.267
0.11522	0.050038	10.636
0.14233	0.043065	7.4075
0.16944	0.037516	5.0594
0.19656	0.033103	3.2871
0.22367	0.029552	1.9072
0.25078	0.026651	0.80524
0.27789	0.024246	-0.093784
0.305	0.022226	-0.84041

Figure 7a: Optimal Blade Geometry

Building off of this geometry we can also make a plot to show the distribution of the chord and pitch angles to better understand how both of the parameters change throughout the length of the blade. Below is a figure that shows this distribution.

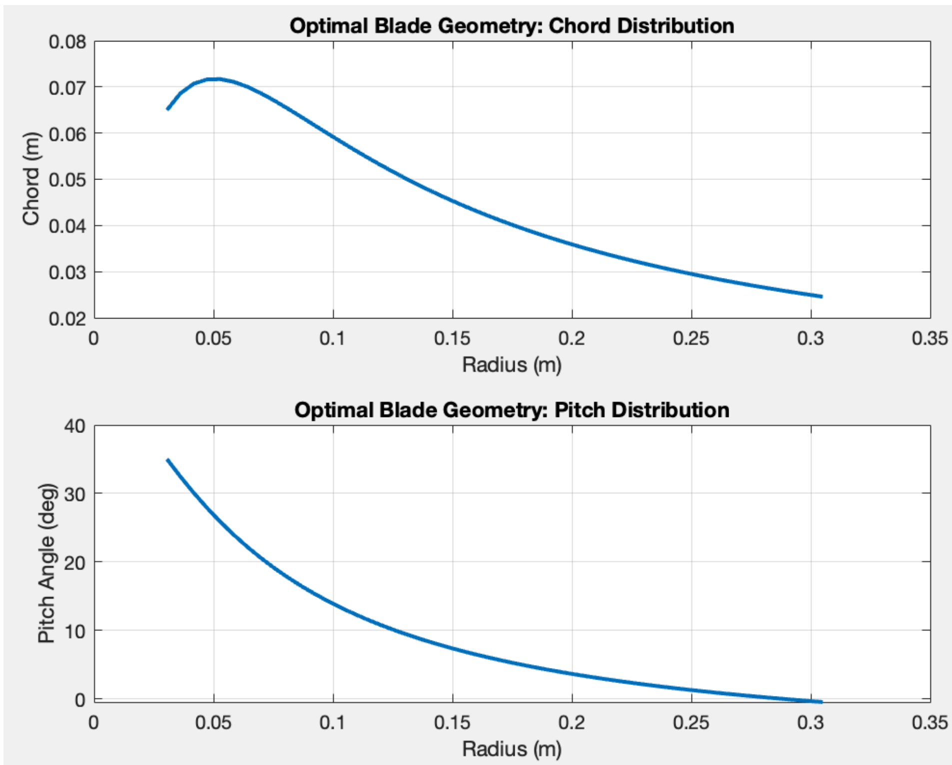


Figure 7b: Blade Geometry Distribution

Power of BEM Design

Again using the script and method above, at the optimal TSR of $\lambda_r = 4.6$, the power coefficient $C_p = 0.436$. Below as a figure is the distribution of the power coefficients with respect to the range of different TSR.

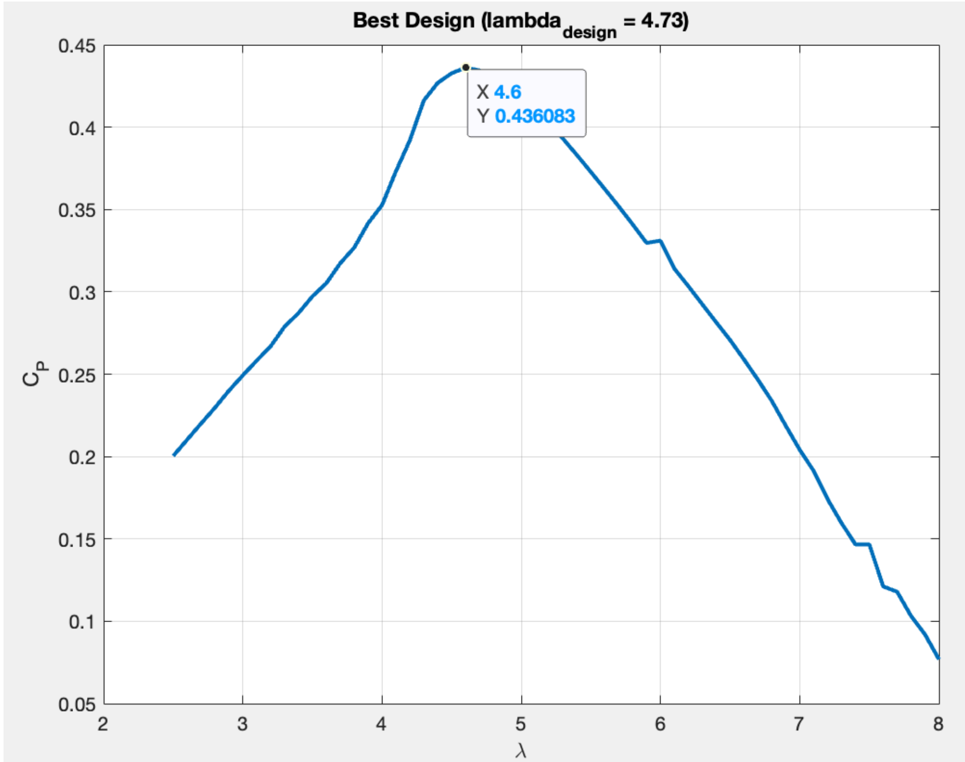


Figure 8: Optimal Blade Geometry Power Coefficient

Using this power coefficient we can also use methods described above to get the suspected power output of the wind turbine for: constant wind speeds, variable wind speeds.

For the constant wind speed assumption, using the definition of the power coefficient we can get the expected power out. From the fact that the power coefficient is the ratio of power transferred to the rotor to the overall power in the air, we can easily calculate the power output using the below method.

$$C_P = \frac{P_{rotor}}{P_{air}} = \frac{P_{rotor}}{\frac{1}{2}\rho A U^3} \Rightarrow P_{rotor} = \frac{1}{2}\rho A U^3 \times C_p$$

From this we got that at a wind speed $U = 6m/s$, $\rho = 1.225kg/m^3$, and $A = \pi R^2 = 0.2922m^2$, we got an expected $P_{rotor} = 17.08W$. It is important to note that this is just accounting for the aerodynamic efficiency of the turbine, so this does not account for mechanical losses in the turbine like the torque friction in the rotor, or other imperfections in the conversion of this work in the rotor into electrical energy. This number should therefore be seen as an upper bound for the blade power coefficient.

Comparison of Standard Turbine and Optimal Design

Comparison of Standard to Optimal Design

To rigorously analyze the differences between the standard design supplied by the class, and my optimal design, I compared the two designs with code, that took in the geometry of both of the design, and then graphed both of their C_p values along a range of different TSR. Below is a figure to visually analyze the differences between the overall shape of either design.

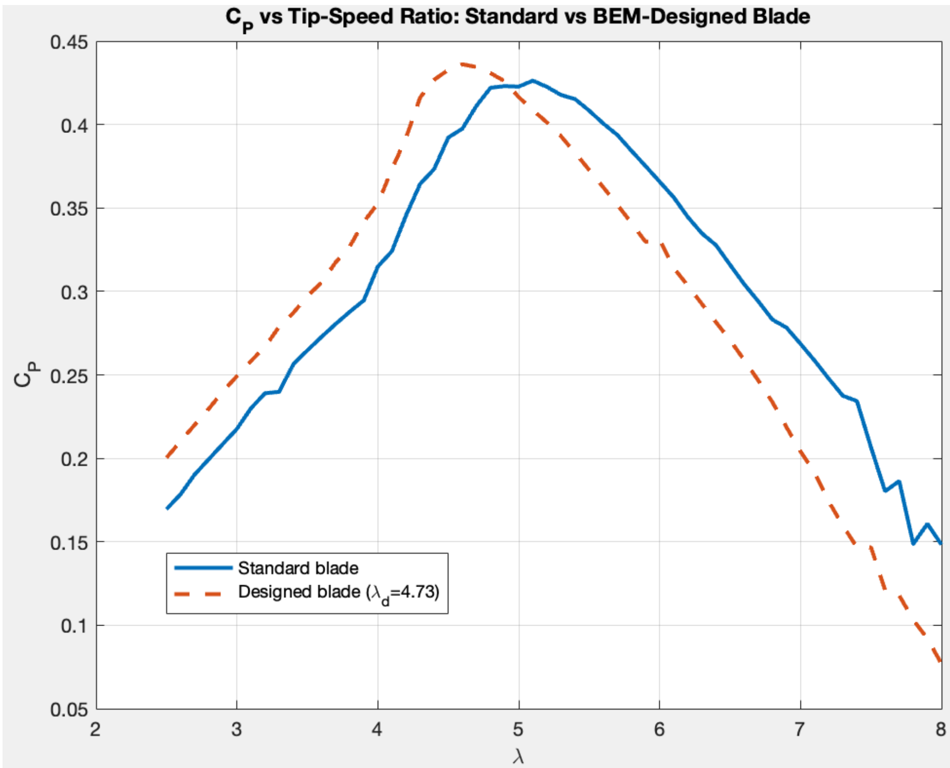


Figure 9a: Standard Blade Compared to Optimal Design Blade

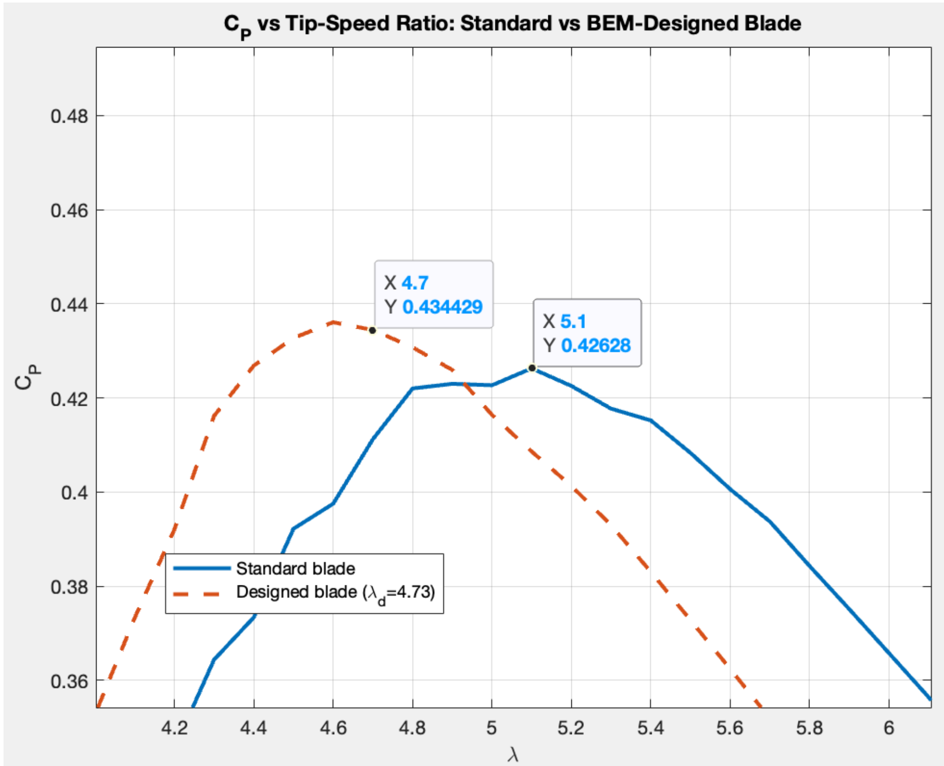


Figure 9b: Zoomed in Blade Comparison

There are many important things to consider when analyzing this graph as it applies to both this report and the overall optimal consideration of this blade.

The first thing to note is that we have successfully designed a blade that has a higher power coefficient, and subsequent higher efficiency in turning wind energy into mechanical useful work. That is to say, if we were to hold all things constant, where we held both blades at their optimal TSR, and they were made of all the same components and materials, my design will have a higher power output compared to the standard design.

Another important thing to recognize is that since the optimal TSR is lower, that means in the end the turbine, if implemented in real world circumstances, will experience less blade erosion. This comes from the principle that real world blade erosion is a result of the fatigue stress on the blade, when put under extreme weather conditions, like hail and rain. The way this connects to the TSR is that the fatigue stress and rate of blade erosion is proportional to the linear speed of the blade at a given point along the span of the blade. If we were to consider a frame of reference of a water droplet or piece of hail falling in the sky, the relative velocity between the piece of hail and the turbine is the speed of hail plus the linear speed of the turbine at that point. In the most extreme case of hail hitting the tip of a turbine, which has the highest linear speed of the turbine, the impulse of the hail would be higher when the tip was moving really fast, such is the case of higher TSR values. All of this is to recognize, although we were not optimizing our turbine for high or low TSR or lower effect of blade erosion, by optimizing our blade in this way we believe our turbine will erode less, which is a happy side effect of the design.

The last thing to note is that since we decided to optimize for a certain TSR, which fueled a lot of our governing equations with regards to determining the geometry of the turbine, we effectively made the turbine specialize in that TSR. In doing so, our range of TSR somewhat dropped off, such that overall our turbine is less efficient at extreme speeds compared to the standard blade counterpart, which is better at handling a wider range of TSR. We can quantitatively analyze this by integrating under the curve formed by the power coefficient with respect to the TSR. By integrating under this curve, which was done using the **trapz** function in matlab, we got that in the range of $TSR = (2.5 - 8)$:

$$C_{p,avg} = \frac{1}{n} \int_{TSR_0}^{TSR_f} C_p d(TSR)$$

Standard Blade: $C_{p,avg} = 0.3034$

Optimal Blade: $C_{p,avg} = 0.2849$

This means across all TSR values, on average, the ‘optimally designed blade’ is roughly 6.5% less efficient than the standard blade. This gives us the notion that because the blade is optimized for a single TSR = 4.76, the generality of the blade is reduced in terms of its range.

To conclude the comparison between the standard blade and the blade designed for optimization of power, we successfully designed a turbine that is better at converting wind energy into mechanical work, by optimizing the pitch and chord distribution for the blade. Through optimizing this power, we were able to generate higher power at a lower TSR λ_r, which subsequently allowed the blade to be less susceptible to blade erosion, and therefore has a longer operating lifespan. But by developing the blade to specialize in this ‘golden-locks’ zone of lower TSR values, we decreased the blades overall efficiency across all TSR values, which by specializing the blade to a certain value, we lost the generality of all values, which consequently binds us to a certain operating range.

Next Steps

Although in this report rigorous analysis of both the Standard blade, using Ansys Fluent CFD, BEM theory, and experimental data, and the optimally designed blade, using BEM theory, was done, there is still a lot more things to be done to increase the fidelity of our design, along with increasing its viability for implementation. In this next section I would like to expand on the next steps going forward in 3 main categories

Design: Now that we have developed an optimal geometry using the BEM method, the next step in the development process would be to build this model into a 3D structure in CAD, both as a singular blade, and also as an entire 3 blade wind turbine (hub included).

Simulation of Fluid Flow: Building off the above **Design** stage of development, one can then use that model of the new blade, and run more fluid flow simulations on it. This would include parameterizing the set up to sweep through the different TSR, and see how that would truly affect the turbine torque and power output. One can also consider improving the quality of the mesh past the point of refinement seen in this report.

Experimentation: After doing further fluid analysis, it is then important to do further testing of the blade in real world turbulent flow conditions. To test this experimentally, one can 3D print a model of the blade, using the previously constructed blade geometry in CAD, then use that in a wind tunnel set up. Using this wind tunnel set up we can empirically learn more about the power output of the blade, its loading and structural failure cases, and the wake velocity and effect on the turbine efficiency.

Structural FEA: Beyond considering the effect of TSR with respect to blade erosion, no structural analysis work was done here, so going forward it would be important to look at the loading cases for many different wind speeds, and ensuring that at a certain wind speed the turbine will have a high factor of safety against yielding stress along the blade and about the rotor hub. With this analysis we can determine if this design is ready for real world implementation or further development is needed. Furthermore, one should consider fatigue stress, and how that would affect the operating lifespan of the turbine alongside the factor of blade erosion.

Vibration Analysis: In addition to the previous point of structural analysis, further analysis can be done with respect to analyzing the vibration effects of this blade in real world environments. This is especially important when suspecting and analyzing the resonance nodes of the system, and ensuring the blades which are subject to the highest stress, won't reach any resonance whistle in normal operating conditions.

Analyzing and Optimizing for Variable Wind Speed: In the real world environment, there is no ideal operating wind speed for turbines, so before implementation there must be analysis done on the blade at different wind speeds, along with further analysis of the blade at a variation of wind speeds. One should consider applying the Weibull distribution to simulate different wind speeds for a certain area, and see the performance metrics of this blade out in the variable wind.

Resources

Sources Used:

Manwell, J. F., et al. *Wind Energy Explained: Theory, Design and Application.* 2nd ed., Wiley, 2009.

"NACA 2414 Airfoil (Re = 50,000, Ncrit = 9)." *AirfoilTools.com*, https://airfoiltools.com. Accessed 16, Oct. 2025.

ANSYS, Inc. *ANSYS Fluent User's Guide.* ANSYS, 2024.

"Wind Blade Analysis for Wind Power Using ANSYS Fluent" *ANSYS Innovation Space*, ANSYS, https://innovationspace.ansys.com/product/wind-blade-analysis-for-wind-power/.

"NACA 2414 Wind Tunnel Data Revised." *NACA 2414 wind tunnel data revised.xlsx*.

Appendix A: Matlab

comp_Cp_w_geometry.m

```
function [TSR_range, Cp_vals, r, c, pitch_deg] = comp_Cp_w_geometry(lambda_design)
   %% 1. Load airfoil data
   data = readmatrix('xf-n2414-il-50000.csv');  % [alpha_deg, Cl, Cd]
   alpha_deg = data(:,1);
   Cl_data    = data(:,2);
   Cd_data    = data(:,3);
   % find best Cl/Cd
   ratio = Cl_data ./ Cd_data;
   [~, i_best] = max(ratio);
   alpha_opt_deg = alpha_deg(i_best);
   Cl_opt         = Cl_data(i_best);
   % Cd_opt       = Cd_data(i_best);  % not strictly needed here
   %% 2. Blade parameters + stations
   R = 0.305;       % rotor radius [m]
   B = 3;           % blades
   N = 50;          % number of radial stations (same as before)
   r_root = 0.10 * R;
   r      = linspace(r_root, R, N);   % row vector
   %% 3. Geometry from lambda_design
   lambda_r = lambda_design * (r / R);
   phi      = (2/3) * atan(1 ./ lambda_r);   % inflow angle [rad]
   % chord distribution
   c = (8 .* pi .* r .* (1 - cos(phi))) / (B * Cl_opt);
   % pitch distribution (what we called twist before)
   pitch_rad = phi' - deg2rad(alpha_opt_deg);  % column vector
   pitch_deg = rad2deg(pitch_rad);
   %% 4. BEM performance sweep (Cp vs TSR)
   rho   = 1.225;      % kg/m^3
   U_inf = 8;          % m/s
   TSR_range = 4:0.1:8;
   Cp_vals   = zeros(size(TSR_range));
   for w = 1:length(TSR_range)
```

```
        lambda = TSR_range(w);
        omega  = lambda * U_inf / R;    % rad/s
        cp_sum = 0;
        for n = 1:N
            r_n     = r(n);
            c_n     = c(n);
            theta_n = deg2rad(pitch_deg(n));        % pitch in radians
            sigma_n = B * c_n / (2*pi*r_n);         % solidity
            a  = 0.3;
            ap = 0.0;
            for iter = 1:100
                U_ax  = U_inf * (1 - a);
                U_tan = omega * r_n * (1 + ap);
                phi_n = atan2(U_ax, U_tan);
                alpha_loc_rad = phi_n - theta_n;
                alpha_loc_deg = rad2deg(alpha_loc_rad);
                % clamp alpha to airfoil data range
                alpha_loc_deg = max(min(alpha_loc_deg, max(alpha_deg)), min(alpha_deg));
                % lookup Cl, Cd
                Cl = interp1(alpha_deg, Cl_data, alpha_loc_deg, 'linear');
                Cd = interp1(alpha_deg, Cd_data, alpha_loc_deg, 'linear');
                % normal / tangential coeffs
                Cn = Cl*cos(phi_n) + Cd*sin(phi_n);
                Ct = Cl*sin(phi_n) - Cd*cos(phi_n);
                % tip-loss factor (same form as your friend's)
                F = (2/pi) * acos( exp( (-0.5*B*(1 - r_n/R)) / ((r_n/R)*max(sin(phi_n),1e-6)) ) );
                F = max(F, 1e-3);
                % local thrust coefficient
                C_Tloc = sigma_n * (1 - a)^2 * Cn / max(sin(phi_n)^2, 1e-8);
                % update a
                if C_Tloc < 0.96
                    a_new = 1 / (1 + (4*F*sin(phi_n)^2)/(sigma_n*Cn));
                else
                    a_new = (1/F)*(0.143 + sqrt(0.0203 - 0.6427*(0.889 - C_Tloc)));
                end
                % update a'
                ap_new = 1 / ((4*F*sin(phi_n)*cos(phi_n))/(sigma_n*Ct + 1e-8) - 1);
                % convergence
                if abs(a_new - a) < 1e-5 && abs(ap_new - ap) < 1e-5
                    a  = a_new;
                    ap = ap_new;
                    break;
                end
                % relaxation
                damp = 0.5;
                a  = damp*a_new  + (1-damp)*a;
                ap = damp*ap_new + (1-damp)*ap;
            end
            % Cp integrand term (Eq. 3.136-ish)
            lambda_r_loc = omega * r_n / U_inf;
            term = F * sin(phi_n)^2 * ...
                   (cos(phi_n) - lambda_r_loc*sin(phi_n)) * ...
                   (sin(phi_n) + lambda_r_loc*cos(phi_n)) * ...
                   (1 - (Cd/Cl)*cot(phi_n)) * ...
                   lambda_r_loc^2;
            cp_sum = cp_sum + term;
        end
        Cp_vals(w) = (8 / (lambda * N)) * cp_sum;
    end
end
```

optimal_design.m

```
function [best_lambda] = optimal_design
clear; clc;
% range of design TSRs to try
lambda_design_list = 3.5:0.01:5.5;
bestCp        = -inf;
best_lambda   = NaN;
best_geom_r   = [];
best_geom_c   = [];
best_pitch    = [];
best_TSR      = [];
best_Cp_curve = [];
for k = 1:length(lambda_design_list)
   lambda_des = lambda_design_list(k);
   [TSR_range, Cp_vals, r, c, pitch_deg] = comp_Cp_w_geometry(lambda_des);
   maxCp_here = max(Cp_vals);
   fprintf('lambda_design = %.2f --> max Cp = %.4f\n', lambda_des, maxCp_here);
   if maxCp_here > bestCp
       bestCp        = maxCp_here;
       best_lambda   = lambda_des;
       best_geom_r   = r;
       best_geom_c   = c;
       best_pitch    = pitch_deg;
       best_TSR      = TSR_range;
       best_Cp_curve = Cp_vals;
   end
end
fprintf('\nBEST DESIGN:\n');
fprintf('  lambda_design = %.2f\n', best_lambda);
fprintf('  max Cp        = %.4f\n', bestCp);
% Show geometry for best design
disp('Geometry for best design:')
disp(table(best_geom_r', best_geom_c', best_pitch, ...
   'VariableNames', {'Radius_m','Chord_m','Pitch_deg'}))
% Plot Cp curve for best design
figure;
plot(best_TSR, best_Cp_curve, 'LineWidth', 2);
grid on;
xlabel('\lambda'); ylabel('C_P');
title(sprintf('Best Design (lambda_{design} = %.2f)', best_lambda));
end
```

comparison.m

```
clear; clc; close all;
% Run BEM for standard blade
[TSR_std, Cp_std] = standard_blade_lambda();    % make this function RETURN data
% Run BEM for designed blade
[best_lambda] = optimal_design();
lambda_design = best_lambda;   % or whatever you chose
[TSR_opt, Cp_opt, r_opt, c_opt, pitch_opt] = comp_Cp_w_geometry(lambda_design);
% Plot both on the same figure
figure;
plot(TSR_std, Cp_std, 'LineWidth', 2); hold on;
plot(TSR_opt, Cp_opt, '--', 'LineWidth', 2);
grid on;
xlabel('\lambda'); ylabel('C_P');
legend('Standard blade', sprintf('Designed blade (\\lambda_d=%.1f)', lambda_design), ...
      'Location', 'best');
title('C_P vs Tip-Speed Ratio: Standard vs BEM-Designed Blade');
```

standard_blade_lambda.m

```
function [TSR_range, Cp_std] = standard_blade_lambda()
data  = readtable('xf-n2414-il-50000.csv');   % alpha [deg], Cl, Cd
alpha_val = data{:,1};
Cl_tab    = data{:,2};
Cd_tab    = data{:,3};
%parms
rho    = 1.225;         % kg/m^3
U_wind = 6;             % m/s (reference speed, only sets scaling)
R      = 0.305;         % rotor radius [m]
B      = 3;             % number of blades
% Distance from blade base  (m)
r_std = [0.025,0.050,0.075,0.100,0.125,0.150,...
   0.175,0.200,0.225,0.250,0.2648];
% Pitch angle  (deg)
pitch_std_deg = [22.27, 16.04, 11.82, 8.67,...
   6.45,4.57, 3.16,2.13,1.27,0.46,0.00];
% Chord length  (m)
chord_std = [0.0686,0.061180471,0.053228,0.046430701,0.040959126,...
   0.036415656, 0.032649655,0.029620432, 0.027106641, 0.024900803, 0.0237];
N = numel(r_std);   % number of blade elements
TSR_range = 2.5:0.1:8;
Cp_std    = zeros(size(TSR_range));
for w = 1:length(TSR_range)
   lambda = TSR_range(w);
   omega  = lambda * U_wind / R;    % rad/s
   cp_sum = 0;
   for n = 1:N
       r_n   = r_std(n);
       c_n   = chord_std(n);
       theta_n = deg2rad(pitch_std_deg(n));  % local pitch [rad]

       lambda_r = omega * r_n / U_wind;
       sigma_n  = B * c_n / (2*pi*r_n);      % local solidity

       % initial guesses for induction factors
       a  = 0.3;
       ap = 0.0;
       tol = 1e-5;

       for iter = 1:100
           U_ax  = U_wind * (1 - a);
           U_tan = omega * r_n * (1 + ap);

           phi = atan2(U_ax, U_tan);      % inflow angle
           phi_deg = rad2deg(phi);

           alpha_deg = phi_deg - pitch_std_deg(n);
           % clamp alpha into polar range
           alpha_clamped = max(min(alpha_deg, max(alpha_val)), min(alpha_val));

           Cl = interp1(alpha_val, Cl_tab, alpha_clamped, 'linear');
           Cd = interp1(alpha_val, Cd_tab, alpha_clamped, 'linear');

           % normal & tangential coeffs
           Cn = Cl*cos(phi) + Cd*sin(phi);
           Ct = Cl*sin(phi) - Cd*cos(phi);

           % Prandtl tip-loss factor
           F = (2/pi)*acos( exp( (-0.5 * B * (1 - r_n/R)) / ...
                              ((r_n/R)*max(sin(phi),1e-6)) ) );
```

```
            F = max(F, 1e-3);

            % local thrust coefficient
            C_Tloc = sigma_n * (1 - a)^2 * Cn / max(sin(phi)^2, 1e-8);

            % update a and a'
            if C_Tloc < 0.96
                a_new  = 1 / ( 1 + (4*F*sin(phi)^2) / (sigma_n * Cn) );
            else
                a_new  = (1/F)*(0.143 + sqrt(0.0203 - 0.6427*(0.889 - C_Tloc)));
            end

            ap_new = 1 / ( (4*F*sin(phi)*cos(phi))/(sigma_n*Ct + 1e-8) - 1 );

            % convergence check
            if abs(a_new - a) < tol && abs(ap_new - ap) < tol
                a  = a_new;
                ap = ap_new;
                break;
            end

            % relaxation
            damp = 0.5;
            a  = damp*a_new  + (1-damp)*a;
            ap = damp*ap_new + (1-damp)*ap;
        end

        % Cp integrand term (Manwell Eq. ~3.136 form)
        phi_final    = phi;
        Cl_final     = Cl;
        Cd_final     = Cd;
        F_final      = F;

        add_term = F_final * sin(phi_final)^2 * ...
                   (cos(phi_final) - lambda_r*sin(phi_final)) * ...
                   (sin(phi_final) + lambda_r*cos(phi_final)) * ...
                   (1 - (Cd_final/Cl_final)*cot(phi_final)) * ...
                   lambda_r^2;

        cp_sum = cp_sum + add_term;
    end
    Cp_std(w) = (8/(lambda*N)) * cp_sum;
end
figure(1);
plot(TSR_range, Cp_std, 'LineWidth', 2);
grid on;
xlabel('\lambda'); ylabel('C_P');
title('Standard Blade: C_P vs Tip-Speed Ratio');
[CP_max_std, idx_max] = max(Cp_std);
lambda_peak_std = TSR_range(idx_max);
fprintf('STANDARD blade: max Cp = %.4f at lambda = %.2f\n', CP_max_std, lambda_peak_std);
end
```