MAE 4021 Wind Power

Senior Design Project

Sebastian Mostek

Sibley School of Mechanical and Aerospace Engineering
Cornell University
Fall 2025

Contents

1 Executive Summary **2**

2 Introduction **3**

3 CFD wind turbine simulation **4**

3.1 Overview 4

3.2 Procedure 5

4 Derivation of Mathematical Model **6**

4.1 Flow around a turbine with known geometry 6

4.2 Designing a Turbine 10

5 Comparison of the blade performance **11**

5.1 Reference Data in Depth 11

5.2 Ansys Data in Depth 14

5.3 Matlab Data in Depth 15

5.4 Performance of New Design 17

6 Conclusion **20**

Appendices **21**

A Blade Design Algorithm **21**

B Blade Testing Algorithm **21**

1 Executive Summary

What are the desired function(s) of your design?
The design is a wind turbine in specific, but more importantly it is also framework for designing and evaluating the performance of wind turbine blades in general. The ultimate design objective of any wind turbine is simply to extract as much energy from the wind as possible, or, more precisely, to maximize the coefficient of power. The desired function of the testing framework is to provide with minimal computational overhead a reasonably accurate prediction of that coefficient power given only the blade geometry, airfoil characteristics, and operating conditions.

What constraints related to the main function(s) must your design satisfy?
The most significant design constraint is scale. Part of the purpose of this report is to compare computed performance metrics with those determined experimentally. The physical turbine for which data was collected had a radius of $0.305m$, and so the same is true of its digital twin. Another recurring constraint was the availability (or lack thereof) of airfoil performance data, particularly at extreme angles of attack.

What are the performance objectives of your design? (Give quantitative metrics as much as possible).
The principal performance metric is C_P, the coefficient of power, which is to be evaluated via multiple methods at multiple external conditions. The algorithms I designed also return two different types of error to allow for easy consideration of the reliability of the value of C_P it reports

What alternative design concepts were considered?
The assignment did not leave much room for creative interpretation, and the design decisions about the improved blade (airfoil shape and tip-speed ratio) were locked in pretty early and didn't change. While not a design aspect *per se*, the blade design and testing algorithms went through several stages of improvement that usually required conceptual overhauls.

What analyses were used to select among these alternative design concepts?
The portion with the greatest freedom of choice was the selection of an airfoil for the purposes of designing my own turbine blade. For this, I performed a non-exhaustive search for low-reynolds number airfoils with the greatest maximum lift-to-drag ratio. A post-hoc analysis of how the turbine would change when given a different design value of λ was performed.

What industry or society standards were used to inform or evaluate your design?
For the Computational Fluid Dynamics (CFD) portion of the report, Ansys Fluent was used. For the design portion, much of the derivation of the model followed standard Blade Element Momentum (BEM) theory, though with a distinct addition in the form of a novel form of energy accounting. In terms of evaluation, the Coefficient of Power was used, as other industry standard metrics like the Availability or the Power Generation Ratio rely on site-specific factors that were not taken into account in this design.

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.

1. CS 1112 Intro Computing Using MATLAB was critical given the reliance of this project on Matlab

2. MAE 3230 How Fluid Machines Work was useful as the whole of the mathematical modeling portion requires an understanding of fluid dynamics

3. MAE 2030 Dynamics and MAE 4730 Intermediate Dynamics were of indirect utility given the vector basis transformations used and the resulting fictitious forces.

Surprisingly that's really it. Several courses come to mind that would be relevant to wind power as a whole: system dynamics; circuits; material mechanics; probability in some cases; a little bit of heat transfer perhaps. None of these skills are really exercised by this assignment, which, with the exception of MATLAB literacy, could reasonably be performed using only the skills and mathematics learned within this course.

Even the classes I list as being relevant to this project are only tenuously connected, as for the whole of my academic career I have blazed my own intellectual path parallel to yet distinct from that of the actual course curriculum, such as how I learned Fluid Dynamics not so much from the course but rather by teaching myself Continuum Mechanics on the side. Similarly, my keen familiarity with MATLAB has been developed much more through the eternity spent with my independent research than anything I learned in CS 1112. The listed classes might be relevant to this project conceptually, but not, for me, practically. This mindset of always learning things in my own way and always to a far deeper level than strictly necessary has served me well in spite of the immeasurable time it has absorbed. And it is thanks to this mindset that I have the confidence to assert that my unique mathematical model as developed in this paper is superior to that presented in the course textbook.

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?
It's impossible to know for certain without a thorough set of wind tunnel experiments on a physical recreation of my blade design, but from the data available it is apparent that my blade design outperforms the given reference blade and reaches power coefficients that would be respectable in industry.

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?
I doubt this specific design will have any meaningful direct impact, principally because its primary goal is to extract power, yet it is constrained to a scale at which the total amount of power it could extract is too small to be meaningful. Wind power is so clearly out-competed by solar in low-power applications that there is little hope of any turbine designed with a radius of less than a meter ever seeing real usage. Nevertheless, my derivation of the mathematical model of the turbine and the resulting form of the objective function is novel and effective. As such, I hope that this improved optimization method can find its way into larger scale turbines.

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 bulk of the creative portion of the design is a set of MATLAB scripts used to design, visualize, and evaluate a turbine blade. Nothing physical was ever created for this project, though a Ansys project file was created and used.

Describe each student's role in the design project if it was a group project.
This project was individual; everything in this paper is my own work.

2 Introduction

Wind power is on track to become one of the most important sources of electricity as global energy production decarbonizes [2]. With renewables having become cost competitive against traditional forms of energy generation, the market for wind power is expected to grow substantially [1].

As the industry continues to mature, design expectations increase and the need to develop ever more efficient turbines strengthens. Of critical importance to this aim is the design of the turbine blades, with novel design features like trailing-edge serrations beginning to be tested [3]. On account of the exceedingly high capital costs of testing novel designs at scale, the industry is heavily dependent on Computational Fluid Dynamics (CFD) software such as Ansys Fluent. These proprietary systems allow for complex fluid dynamical systems to be simulated and analyzed. However, the computational cost of simulation is intense, with hundreds of computer hours being spent on simulations between design iterations [7].

In contrast, theoretical models represent both a source of much less computationally expensive design feedback, and also offer insights into the most important variables affecting the general system. However, in practice, mathematical models walk a fine line between accuracy and solvability, as every simplifying assumption reduces both the computational load but also the model's applicability.

This paper begins with an overview of the procedure of a simple CFD simulation of the torque on a wind turbine blade using Ansys Fluent. Section 4 then develops a mathematical model and a pair of corresponding algorithms to test a wind turbine with a known shape and to design a wind turbine from a small set of design parameters. Section 5 compares the results of these different procedures to data from wind tunnel experiments, evaluating both the benefits and tradeoffs of each method and the potential improvements accessible through the design algorithm.

3 CFD wind turbine simulation

3.1 Overview

Computational Fluid Dynamics (CFD) is a powerful yet intricate tool for determining the complex interactions between fluids and solids. Ansys Fluent is a commercially available CFD software used extensively in industry and research, and is the tool used for the CFD simulations in this report.

The operational principal of CFD is that partial differential equations (PDEs) can be converted into either a set of coupled time-dependent ordinary differential equations (ODEs) or, for steady state systems, a system of linear algebraic equations. This is possible by discretizing the region of interest and approximating differential operators like ∇ in terms of the differences of values between these discrete points, called cells. By converting the PDEs into a linear system of equations, the tools of linear algebra can be used to solve for the relevant unknowns (velocity components, pressure, etc.) at each point. Of course, this is extremely computationally expensive, as an accurate estimation of the differential operators requires a very fine mesh, so it is not uncommon for the number of cells to reach the hundreds of thousands and, in the case of time-dependent systems, for the computation time to be measured in tens of hours.

The simulation presented here will come nowhere near such extremes as several assumptions have been made to avoid unnecessary complexity. Nevertheless, the time to calculate the solution from initialization was on the order of 10-15 minutes using 4 Intel i5 CPU cores, with subsequent calculations taking about 1 minute.

The equations Ansys will be solving are the fluid dynamical forms of conservation of mass and momentum, known as the Continuity and Equilibrium Equations:

$$\begin{aligned} \frac{D\rho}{Dt} + \rho\nabla\cdot\vec{u} = 0 \\ \frac{D}{Dt}\rho\vec{u} = \nabla\cdot\boldsymbol{\sigma} + \vec{f} \end{aligned} \tag{1}$$

where ρ is density, $\vec{u}$ is the flow velocity, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, and $\vec{f}$ includes external body forces such as gravity. In addition,

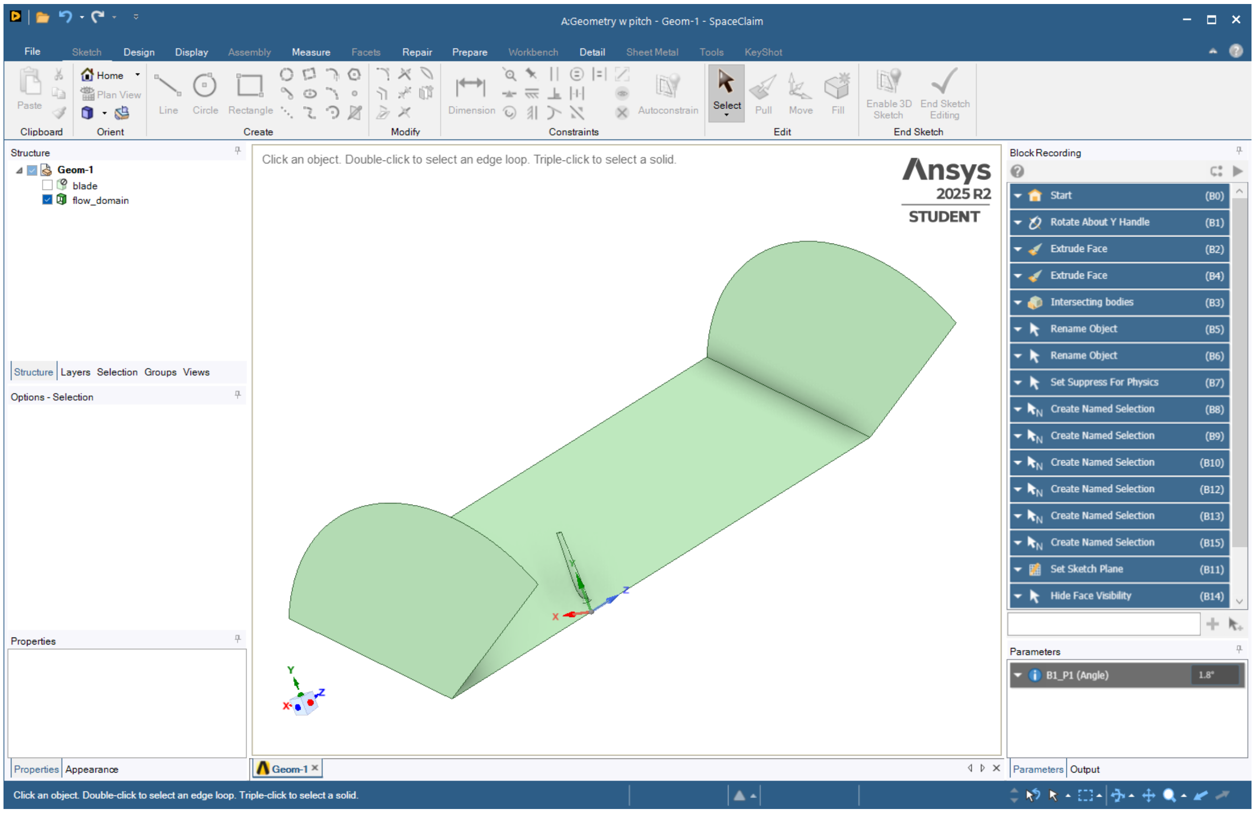
$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{u}\cdot\nabla \tag{2}$$

is the material derivative, which can be thought of as simply a total derivative $\frac{d}{dt}$ where the variation in position is taken to be equal to the local flow velocity.

A few additional assumptions are required in order to solve these equations. First, ρ is taken to be a uniform constant $\rho_{air} = 1.225\,kg\cdot m^{-3}$, which is unlikely to cause any issues unless the flow velocity manages to reach Mach numbers above about 0.3, in which case there would be other more significant concerns. To remove time-dependence from the equations, they are solved within a reference frame which rotates with the blade at a constant angular velocity $\vec{\Omega} = \Omega\hat{z}$ thus introducing the fictitious Coriolis and centripetal forces. Lastly, the Cauchy stress tensor is separated into its component hydrostatic and deviatoric components. The result of all these changes is

$$\begin{aligned} \nabla\cdot\vec{u} = 0 \\ \rho\vec{u}\cdot\nabla\vec{u} = -\nabla p + \nabla\cdot\boldsymbol{\tau} - \rho(2\vec{\Omega}\times\vec{u} + \vec{\Omega}\times\vec{\Omega}\times\vec{r}) \end{aligned} \tag{3}$$

where p is pressure and $\vec{r}$ is the position vector. This equation presents a problem, however: The typical Navier-Stokes Equation results from taking the Equilibrium Equation and applying a constitutive equation relating the deviatoric stress $\boldsymbol{\tau}$ to the flow velocity, resulting in a viscous force term $\mu\nabla^2\vec{u}$. However, the turbulence that arises from this viscosity is inherently time-dependent and exists at a very small scale. But there is a workaround: Wilcox [9] designed a model relating the deviatoric stress to the turbulence kinetic



Click an object. Double-click to select an edge loop. Triple-click to select a solid.

B1_P1 (Angle) 1.8°

Click an object. Double-click to select an edge loop. Triple-click to select a solid.

Figure 1: Fluid domain geometry as viewed in Ansys SpaceClaim just prior to moving to Fluent for meshing. The top face and one of the interior side faces are made invisible to show the position and size of the blade within the domain. Note the Blocks menu on the right with the parameterized rotation step.

energy k and its rate of rate of dissipation ω through a pair of differential equations. This k-ω model, whose details are far beyond the scope of this paper, allows for the velocity field to be solved in steady-state while still capturing the effects of viscosity. For the purposes of this paper, Ansys' GEKO algorithm for this model was used under default parameters.

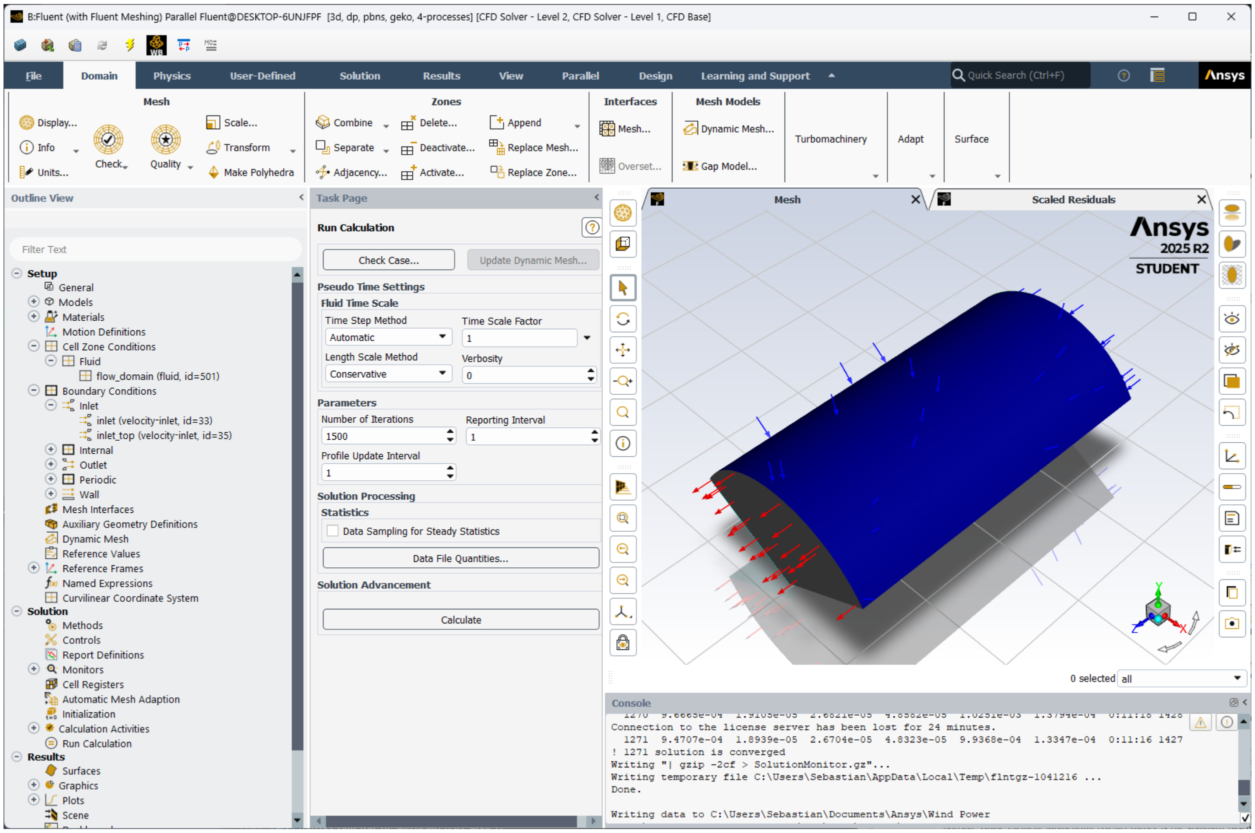
3.2 Procedure

For simplicity, instead of simulating the flow around three blades, a fluid domain consisting of a third of a cylinder surrounding a single blade was used as it is assumed that the flow around each of the three blades in the turbine are equivalent.

The bulk of the process of setting up the simulation is detailed in Reference [8]. As such, this description focuses on those parts which differ from the tutorial. The geometry of the blade was provided by the instructor in a `.wbpz` Ansys project file with a blade pitch angle of about 4 degrees and meshing settings already prepared.

An unexpected but necessary first step was the need to translate the blade slightly so that its rotation axis aligned with the global Z-axis. This was done by measuring the vector difference between the origin and the center of the blade's circular base. Once this was corrected, the pitch angle of the tip was measured and set to zero by counter-rotating the blade.

Given the ultimate objective of simulating flow conditions at a wide range of incoming wind speeds, blade pitch angles, and tip speed ratios, it proved very useful to parameterize the pitch angle so that the resulting change in the geometry of the fluid domain would be automatically calculated. Therefore, at this point the Blocks feature of SpaceClaim was activated and an arbitrary rotation about the global Z was performed. The value of this rotation was converted to a parameter, and, for the first set of simulations, set to zero.



Run Calculation

Pseudo Time Settings

Fluid Time Scale

Time Step Method: Automatic
Time Scale Factor: 1
Length Scale Method: Conservative
Verbosity: 0

Parameters

Number of Iterations: 1500
Reporting Interval: 1
Profile Update Interval: 1

Solution Processing

Statistics

□ Data Sampling for Steady Statistics

Solution Advancement

```
  1270  9.6663e-04  1.9105e-05  2.6821e-05  4.8582e-05  1.0251e-03  1.3794e-04  0:11:18 1428
Connection to the license server has been lost for 24 minutes.
  1271  9.4707e-04  1.8939e-05  2.6704e-05  4.8323e-05  9.9368e-04  1.3347e-04  0:11:16 1427
! 1271 solution is converged
Writing "| gzip -2cf > SolutionMonitor.gz"...
Writing temporary file C:\Users\Sebastian\AppData\Local\Temp\flntgz-1041216 ...
Done.

Writing data to C:\Users\Sebastian\Documents\Ansys\Wind Power
```

Figure 2: View of Ansys Fluent after all boundary conditions are set and a test is ready to be run

From there, the setup followed the tutorial closely: drawing the fluid domain around the blade then cutting the blade out; defining each face as a separate set for the purposes of boundary conditions; calculating a mesh with increased precision around the blade wall.

The last step before the simulation can be run is to set the boundary conditions: the two faces on the interior of the cylinder wedge were set to be periodic, thereby ensuring that the solution would be properly rotationally symmetric; the outlet gauge pressure was set to zero; and the slip conditions over the blade wall were set. Finally, the wind speed of the inlets and the rotation rate of the whole fluid domain were set.

The data collection procedure was a compromise between thoroughness and time-intensiveness. Under the expectation that the coefficient of power should depend primarily on the ratio Ω/u_0 (with u_0 being the ambient windspeed), data was collected at two windspeeds: one fine sampling $\Omega \in \{60, 70, \ldots, 140\}\, rad/s$ at $u_0 = 5\, m/s$; then a coarse sampling $\Omega \in \{60, 80, 100, \ldots, 160\}\, rad/s$ at $u_0 = 6\, m/s$. These values sample the range of tip-sped-ratios $\lambda = \frac{\Omega R}{u_0} \in [4, 8]$ well for the given radius $R = 0.305\, m$ This process was then repeated to get data at pitch angles of $\theta_{p0} \in \{-1.8, -1, 0, 1, 2\}^\circ$ The value of -1.8° was chosen because, for unknown reasons, the parameterization failed at -2°; it is expected that the difference in output resulting from this change is negligible.

A detailed analysis of the data which resulted from this procedure is saved for Section 5, but Figure (3) shows a representative example result of the simulation in the form of a pressure contour near the blade tip.

4 Derivation of Mathematical Model

4.1 Flow around a turbine with known geometry

The simplest fluid mechanical model that can still produce useful insights into the turbine is that of an actuator disk split into concentric annular rings as shown in Figure (4).

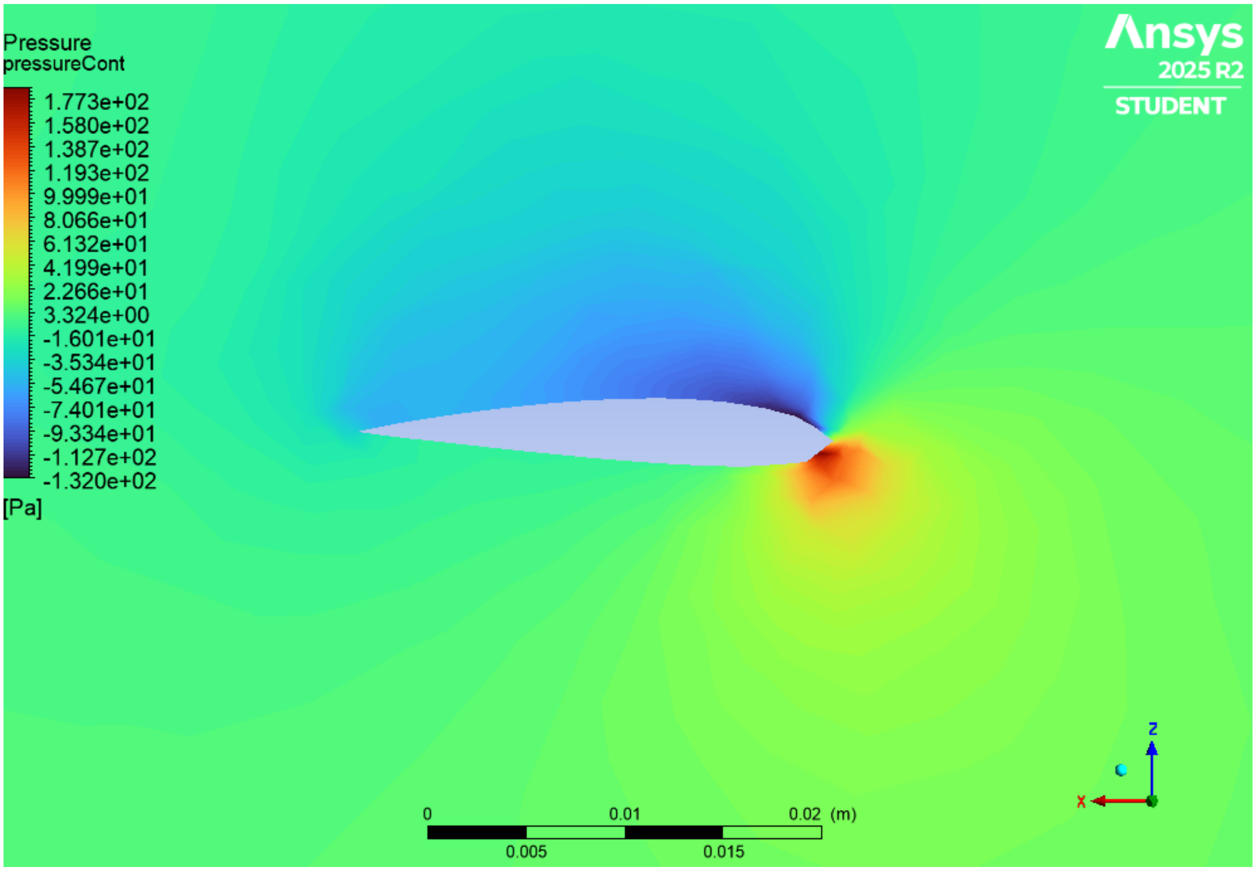


Figure 3: Pressure distribution near the tip, $r = 0.3$, for the conditions $\theta_{p0} = 1°$, $u_0 = 5\,m/s$, and $\Omega = 140\,rad/s$.

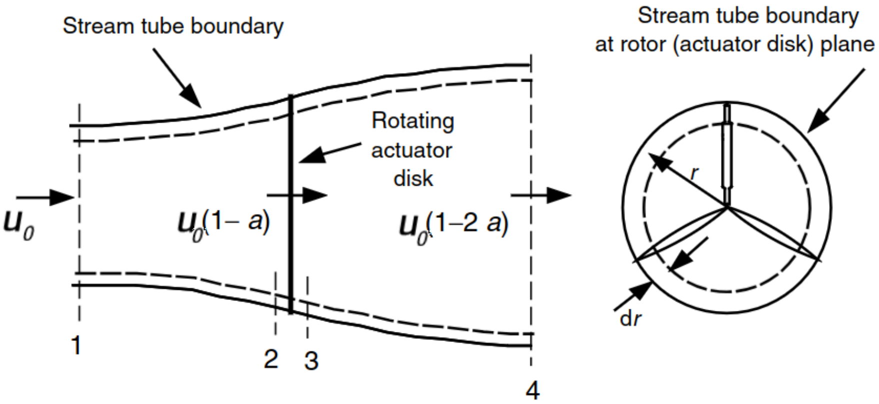


Figure 4: Actuator disk wind turbine model that accounts for wake rotation. A uniform wind profile with speed u_0 is slowed by a factor of a at the plane of the blade. The blade rotates at an angular speed Ω, which induces each annular segment of the plane to rotate at a rate $a'\Omega$ in the opposite direction. Image adapted from Reference [4]

From a consideration of linear momentum on an annular segment of the rotorplane, we can develop two different equations for the thrust perpendicular to that plane

$$\begin{aligned} dF_N &= F_1\,\frac{1}{2}\rho u_0^2(4a(1-a))\cdot 2\pi r dr \\ dF_N &= F_1\,\frac{1}{2}\rho(\Omega r)^2(4a'(1+a'))\cdot 2\pi r dr \end{aligned} \tag{4}$$

where F_1 is an drag factor that accounts for the change in the blade force caused by wake drag, which will be important later. By comparing these two expressions, we define the local speed ratio

$$\lambda_r^2 = \left(\frac{\Omega r}{u_o}\right)^2 = \frac{a(1-a)}{a'(1+a')} \tag{5}$$

Meanwhile, an angular momentum balance on an annular segment yields

$$dQ = F_2 \cdot 4a'(1-a)\pi\rho u_0 \Omega r^3 dr \tag{6}$$

where F_2 is a different drag factor between 0 and 1 that accounts for reductions in torque caused by drag. These equations are not yet useful because neither the induction factors a and a' nor the drag factors F_1 and F_2 are determined. In order to constrain these unknowns, a separate model of the forces on the turbine must be developed, for which we turn to Blade Element Momentum (BEM) theory. All of the core components of the BEM model are shown in Figure (5).

From the trigonometry of the velocity triangle

$$\tan\varphi = \frac{1-a}{\lambda_r(1+a')} \tag{7}$$

which, in combination with Equation (5) allows for a and a' to be expressed solely in terms of φ:

$$\begin{aligned} a &= \cos\varphi(\cos\varphi - \lambda_r\sin\varphi) \\ a'\lambda_r &= \sin\varphi(\cos\varphi - \lambda_r\sin\varphi) \end{aligned} \tag{8}$$

The geometry of the wind triangle also yields

$$\left(\frac{u_{eff}}{u_0}\right)^2 = (1-a)^2 + \lambda_r^2(1+a')^2 = (\lambda_r\cos\varphi + \sin\varphi)^2 \tag{9}$$

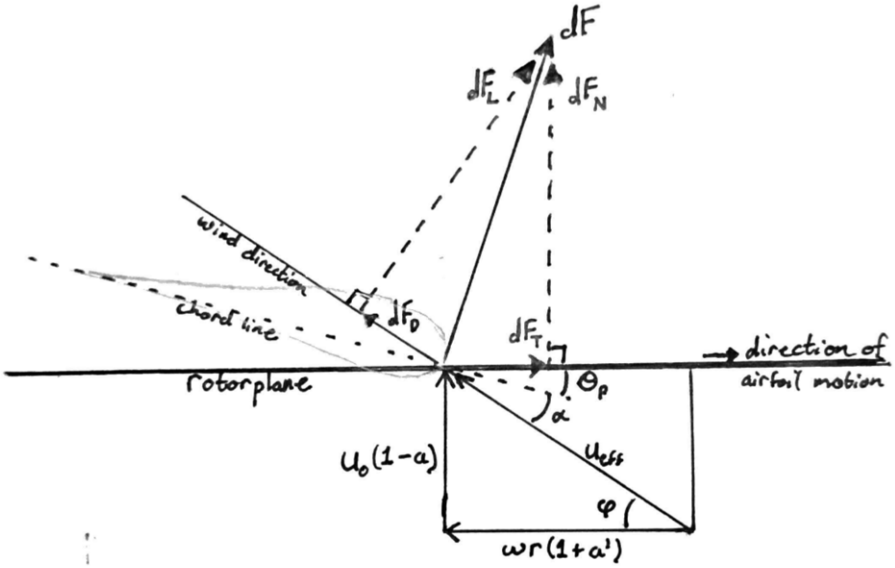


Figure 5: Cross Section of turbine blade. The lift and drag coefficients, C_L and C_D, are empirically evaluated functions of the angle of attack α and the Reynolds Number Re. The force magnitudes are related to these coefficients as $dF = C\frac{1}{2}\rho u_0^2 c dr$ where c is the chord length. The amount of time required to draw this compared to the quality makes me think it would have been cost effective to learn how to make it digitally.

Meanwhile, considering the force vectors, the normal force perpendicular to the rotorplane dF_N and the power-producing torque $dQ = r\,dF_T$ for a turbine with B blades can be readily determined from Figure (5) as well as the definitions of the coefficients of lift and drag C_L and C_D

$$\begin{aligned} dF_N &= B \cdot C_N \tfrac{1}{2}\rho u_{eff}^2\, c\, dr \\ dQ &= B \cdot r \cdot C_T \tfrac{1}{2}\rho u_{eff}^2\, c\, dr \\ \text{where} \qquad C_T &= C_L \sin\varphi - C_D \cos\varphi \\ \text{and} \qquad C_N &= C_L \cos\varphi + C_D \sin\varphi \end{aligned} \tag{10}$$

Equating these expressions with those determined from the annular actuator disk — Equations (4) and (6) — reveals

$$\begin{aligned} F_1 &= \frac{\sigma'(C_L + C_D \tan\varphi)\tan(\varphi + \tan^{-1}\lambda_r)}{4\sin\varphi} \\ F_2 &= \frac{\sigma'(C_L - C_D \cot\varphi)\tan(\varphi + \tan^{-1}\lambda_r)}{4\sin\varphi} \end{aligned} \tag{11}$$

where several algebraic steps and trigonometric simplifications have been skipped in the name of brevity. The term $\sigma' = Bc/2\pi r$ is the local solidity, another nondimensional parameter. The significance of these equations is twofold: for one, they demonstrate the need for the axial and rotary drag factors F_1 and F_2 to be considered separately — Reference [4] goes through the same derivations shown here but assuming $F_1 = F_2 = 1$ for the majority of the process, then adds a drag term F afterwards, claiming it allows for the equivalence between the actuator disk model (which otherwise ignores drag) and the BEM model. But Equation (11) clearly shows that F_1 can only equal F_2 if $C_D = 0$, which defeats the purpose of including the drag factor. Secondly, if the blade geometry is taken as a given (i.e. σ' and the pitch angle θ_p are known functions of λ_r), then this system has two equations and three unknowns: F_1, F_2, and φ. The final equation required to fully determine this system is Prandtl's model for tip loss:

$$F_2 = \frac{2}{\pi}\cos^{-1}\left(\exp\left(\tfrac{B}{2}(1 - \tfrac{\lambda}{\lambda_r})\csc\varphi\right)\right) \tag{12}$$

This equation was developed by Prandtl [6] and comes from a convoluted analysis of the flow circulation in a helical wake that is well beyond the scope of this paper.

Now, between Equations (11) and (12), the wind conditions can be fully determined, though this system is too nonlinear to be solved analytically and must be determined through computational optimization, for which I used the MATLAB function `fsolve`. This requires a function that takes a vector of φ values along the blade and uses it to evaluate the difference between the values F_2 as calculated by Equations (11) and (12). The optimizer then varies the input φ vector until the output difference vector is as close to all zeros as possible. One complication in this process is that the values of C_L and C_D are discrete data vectors and so cannot be evaluated at an arbitrary value of $\alpha = \varphi - \theta_p$. I chose to account for this by simply taking the values at the nearest available value of α, as a more elaborate interpolation seemed unlikely to meaningly affect the result. Moreover, of greater significance is the lack of data at extreme values of α: as shown in Figure (9), there is no data on airfoil performance beyond the range of approximately $-10°$ to $+12°$. This informs one of two forms of error reported by the optimization function. The program returns the proportion of the calculated values of φ that correspond to α beyond the range for which lift and drag data exist. The other error is the root mean square of the residuals returned at the end of the optimization's runtime.

Once φ has been determined, the torque on the blade can be calculated using Equation (10). Finally, the coefficient of power can be calculated as the ratio of power produced to power available:

$$C_p = \frac{\int \Omega dQ}{P_{wind}} \tag{13}$$

where $P_{wind} = \frac{1}{2}\rho u_0^3 \pi R^2$ is the total amount of wind power in a free flow over a circular area of radius R.

4.2 Designing a Turbine

If the geometric parameters σ' and θ_p are unknown, then the derivation of the previous section does not provide enough information to constrain them. As such, we must make some simplifications in order to reach a point at which the geometry can be optimized. Referring back to Equation (6), if the drag is ignored ($F_2 = 1$), this expression for torque can be substituted into Equation (13) to get

$$C_P = \frac{8}{\lambda^2} \int_0^{\lambda} a'(1-a)\lambda_r^3 d\lambda_r \tag{14}$$

which is maximized in the case that

$$a' = \frac{1-3a}{4a-1} \tag{15}$$

Plugging Equation (15) into Equation (5) and rearranging gives the following polynomial relation between a and λ_r

$$16a^3 - 24a^2 + (9 - 3\lambda_r^2)a + (\lambda_r^2 - 1) = 0 \tag{16}$$

Although I haven't seen this reported anywhere, since this equation is cubic in a, it can be solved explicitly to give a as a function of λ_r. The procedure for this is tedious and not especially relevant, but the end result is

$$a = \frac{1}{2} - \frac{\sqrt{\lambda_r^2 + 1}}{2} \cos\left(\frac{\pi + \tan^{-1}\lambda_r}{3}\right) \tag{17}$$

This equation[1] can then be reinserted into Equation (14) to numerically evaluate the ideal coefficient of power as a slightly more nuanced function than the simple Betz Limit. However, while this set of equations is what is used in textbooks and research as the upper limit of C_P, a more accurate picture can be gained with little extra effort by simply including the effect of F_2 since it also has a known empirical expression. By expressing F_2, a, and a' in terms of φ, it is very easy to pass an objective function returning the coefficient of power to a numerical optimizer like MATLAB's `fminunc` and get an estimate of the distribution of φ that would maximize the turbine power. Admittedly, the effect is very slight, but there's not much reason to avoid the implementation. From there, considering the definition of dQ given in Equation (10), it is clear

[1] As an aside, the form of Equations (11) and (17) make me think that $\tan^{-1}\lambda_r$ would make for a more useful nondimensional parameter than λ_r.

that to maximize C_P, one must maximize $C_T = C_L \sin\varphi - C_D \cos\varphi$, which is also simple to do numerically. At that point, the design can be finalized by retrieving the pitch angle and local solidity using

$$\begin{aligned} \theta_p &= \varphi - \alpha \\ \sigma' &= \frac{4F_2 \sin^2\varphi}{C_T \tan(\varphi + \tan^{-1}\lambda_r)} \end{aligned} \tag{18}$$

where the the latter expression is a rearrangement of Equation (11). One issue with this method is that it assumes that the Reynolds number is a known constant along the length of the blade. A more complicated optimization process could theoretically take this into account, but in practice the availability (or lack thereof) of data for C_L and C_D at different Reynolds numbers constrains the optimization. A blade of this size ($R \approx 0.3m$) can be expected to have Reynolds number on the order of about 10^4 to 10^5, but *airfoiltools.com* only provides data at discrete values $Re = 5 \cdot 10^4$ and $Re = 10^5$. Luckily, the error caused by this simplification is small as the variability of the lift and drag coefficients with respect to Re is significantly less than its variability with respect to α. Nevertheless the Reynolds number of the final design will be checked to ensure that the results are reasonable.

A more significant caveat of this method is that by skipping much of the math of the previous section, there is no guarantee that the result of the air-turbine interaction will actually result in the calculated optimum distribution of φ. This distribution is assumed and the blade is designed around it. This imprecision is why engineering is a discipline: there is not enough mathematical constraints to fully determine the system so we make guesses and test them until general rules of thumb are determined. For instance, one of the key decisions of using this design algorithm is the selection of a value of λ. Nothing in the math has indicated that this value matters, but it has become part of the wind turbine set of guidelines that a tip-speed ratio near 7 is effective.

It is also worth noting that the optimization just described is slightly non-standard. Typically, one would choose the fixed value of α that maximizes the lift-to-drag ratio C_L/C_D. A comparison of the distribution of α resulting from this method and that resulting from my method of maximizing C_T is shown in Figure (6). The downside of my more complicated method is that it increases the value of dF_N and thus the total bending load on the blade, but such mechanical considerations are beyond the scope of this paper, and at the scale of this turbine there is no reason to suspect this slight increase in bending moment would be significant. Nevertheless, the camber of the airfoil was scaled by 50% when defining the 3D blade geometry in order to ensure that the blade was not dangerously thin — this would doubtless have effects on a real blade but it is not something that is accounted for in thin-airfoil theory.

Finally, to take the design out of the abstract, a specific airfoil must be chosen. Knowing that the small blade radius would limit the Reynolds number, I performed a search on *airfoiltools.com* for an airfoil with the highest maximum value of C_L/C_D at the lowest Reynolds number with data available, namely $Re = 50000$. The winner was the Selig 7075 with a maximum C_L/C_D of 39. The only design parameters left are R, which is fixed at $0.305m$, and λ, which I set at 7. The resulting geometry is shown in Figures (7) and (8)

5 Comparison of the blade performance

In order to fairly compare the different readings provided by the different evaluation methods — CFD simulation versus theoretical model — it is necessary to have a benchmark to measure from. Luckily, the results of wind tunnel tests on a print of the NACA 2414 blade at a range of operating conditions have been provided by the instructor. It is useful to begin with an analysis of this reference data.

5.1 Reference Data in Depth

Central to wind turbine theory and practice is the tip-speed ratio λ. As with most nondimensional numbers, it seeks to generalize the relationship between multiple different variables to reduce the total number of variables that need to be considered. In the case of tip-speed ratio, the model articulated in section 4 indicates that an increase in ambient wind speed coupled with a proportional increase in rotation rate — thereby keeping λ the same — should have no effect on turbine performance. In oppostion to this assumption, there is the wind tunnel data. Figure (10) shows the wind tunnel data disaggregated by blade pitch angle

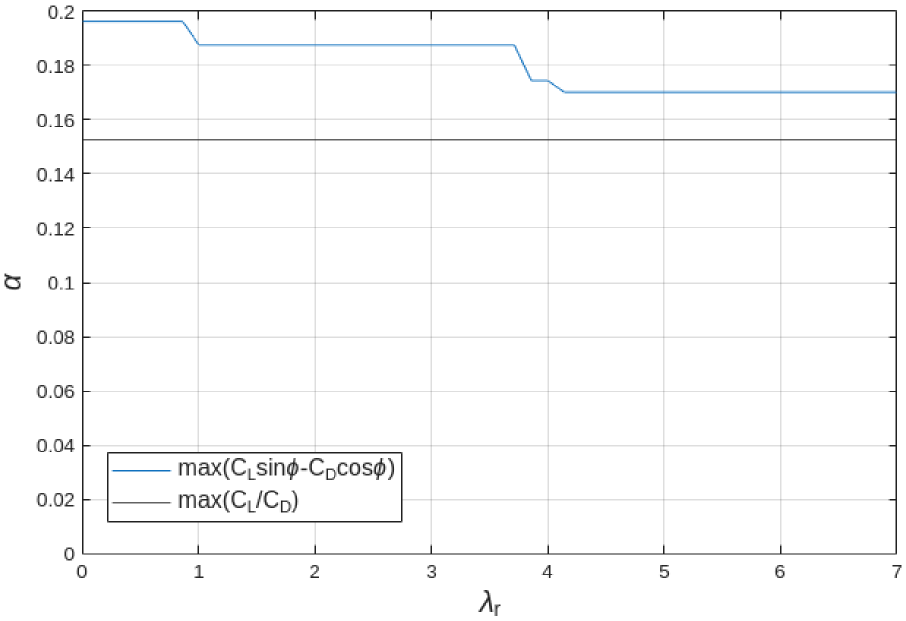


Figure 6: Distribution of angle of attack using two different optimization methods given the data for the s7075 airfoil and a design tip speed ratio $\lambda = 7$. The discontinuity of the blue line is because the data for lift and drag coefficients is discretized.

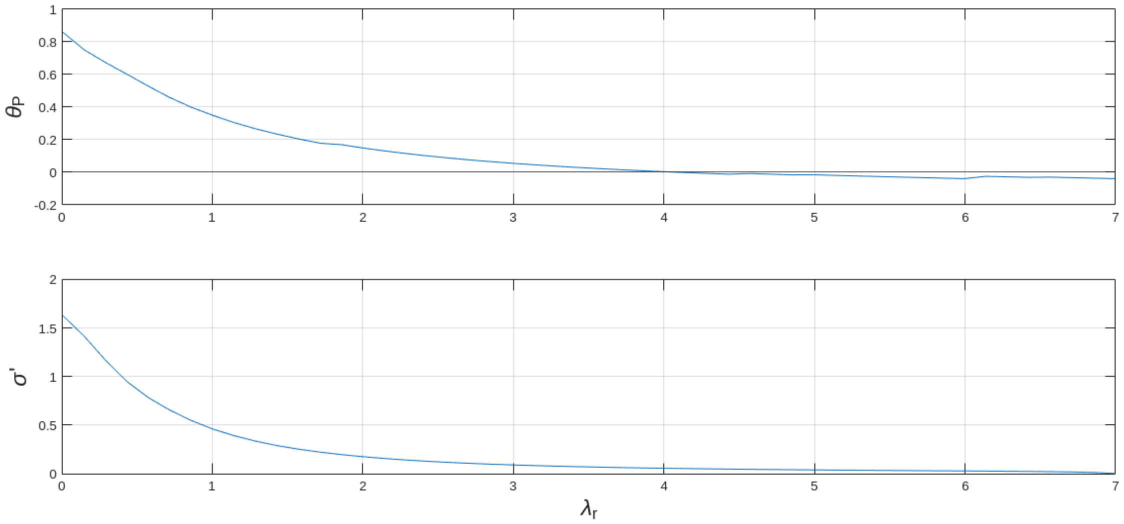


Figure 7: Pitch angle θ_p and chord length c given a design $\lambda = 7$. It is worth noting that the blade pitch angle $\theta_{p0} = \theta_p(\lambda_r = \lambda) = -2.3619°$

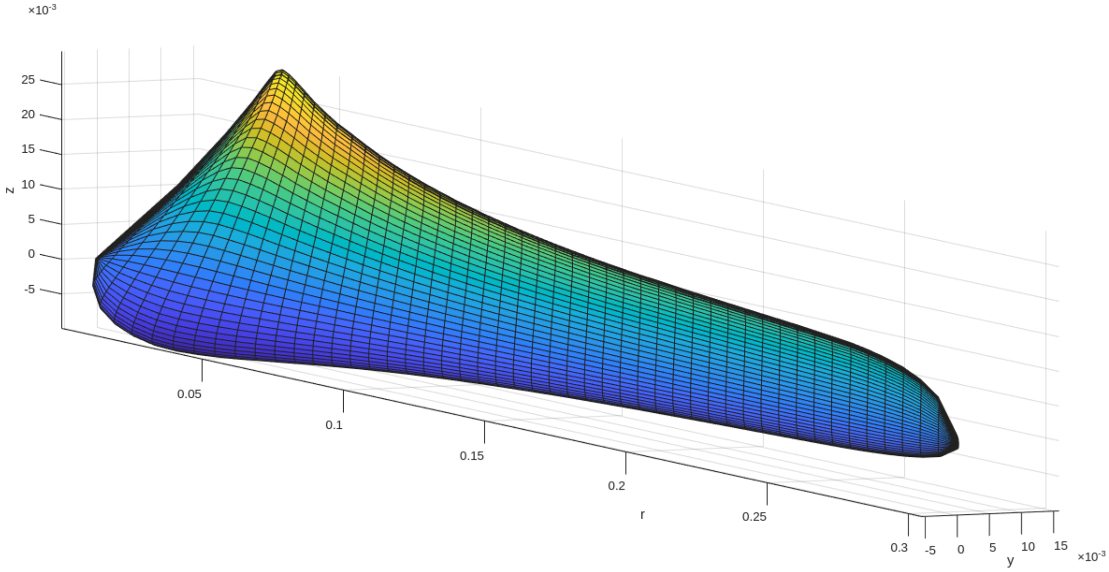


Figure 8: 3D surface of the blade design, Blade is oriented such that the wind is coming from -y and the blade is rotating towards -z.

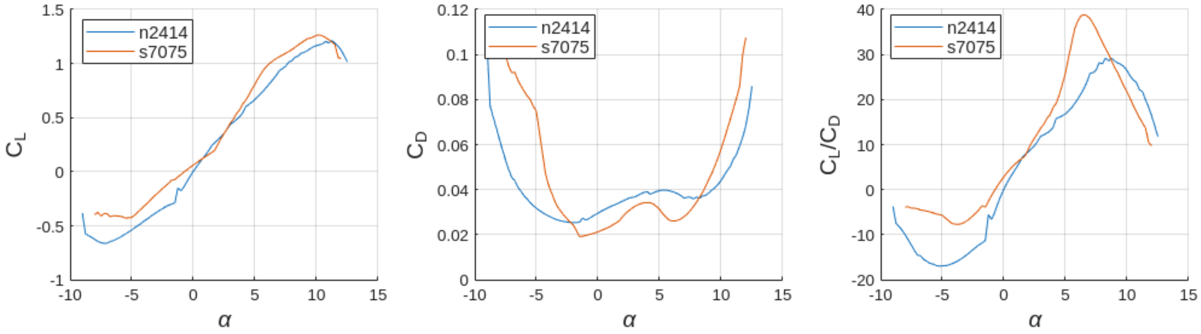


Figure 9: Comparison of the characteristics of the NACA 2414 and Selig 7075 airfoils at $Re = 50000$. Though similar in their overall shape, the slightly greater lift and slightly lower drag of the Selig contributes to a significant peak in the lift to drag ratio near $\alpha = 6°$.

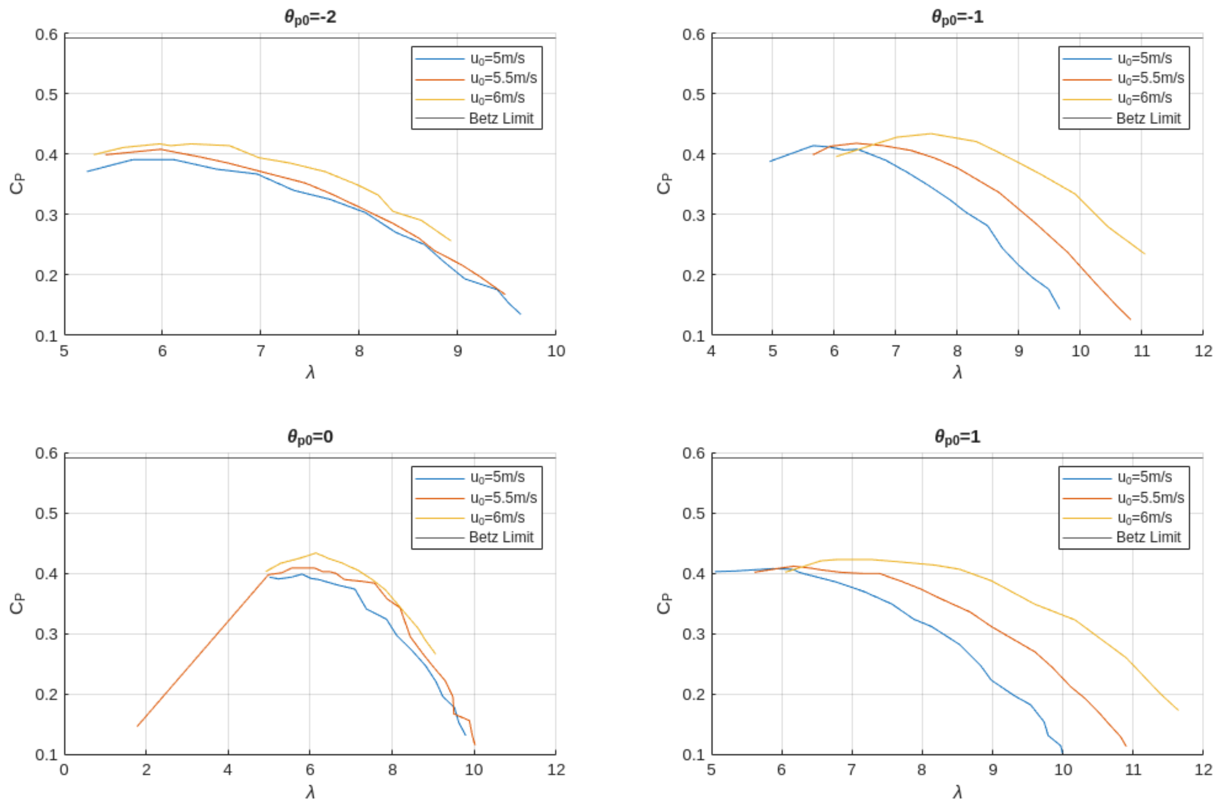


Figure 10: Wind turbine data disaggregated by blade pitch angle θ_{p0} and ambient wind speed u_0. The variability of C_P at a fixed value of λ complicates this data set as a benchmark for further comparison. Angles are in degrees.

and ambient wind speed, and the results are clear: at a fixed value of θ_{p0} and λ, there can exist significant differences in performance. For instance, at a blade pitch angle of 1°, the coefficient of power varies by roughly 30% at $\lambda = 8$ just from the change in wind speed from 5 to 6 m/s, and the disagreement only increases as ω grows.

This variance is likely a product of several factors. For one, there is bound to be variability in the measurement even under nominally equivalent operating conditions: wind speed varies with both space and time in ways that a single measurement cannot reflect; torque transducers and tachometers are imprecise and suffer from hysteresis. Moreover, the model's prescription of the invariability of the system at constant λ does not factor in the effect of Reynolds Number, which is known to affect airfoil performance but which is ignored out of practical necessity. These factors are not sufficient in my mind of explaining the amount of variance measured — nor, for that matter, why at other pitch angles like 0° the variance is so relatively small — but to some extent it doesn't really matter. The data only really starts to reach problematic levels of variance above about $\lambda = 8$, and only then for a couple of pitch angles. Arguably it is of greater inconvenience that there exists so little data at lower values of λ, as the range of tip-speed ratios for testing was assigned as 4 to 8.

The takeaway is ultimately that any comparison between data sets will have to be performed qualitatively rather than quantitatively, as a proper analysis would require a great deal more data.

5.2 Ansys Data in Depth

As detailed in Section 3.2, simulations of the wind turbine were performed over a fine sampling of λ at $u_0 = 5\,m/s$, then again at a coarse sampling at $u_0 = 6\,m/s$. The resulting C_P data at several different pitch angles is shown in Figure (11). An encouraging departure from the reference data is the strong agreement in

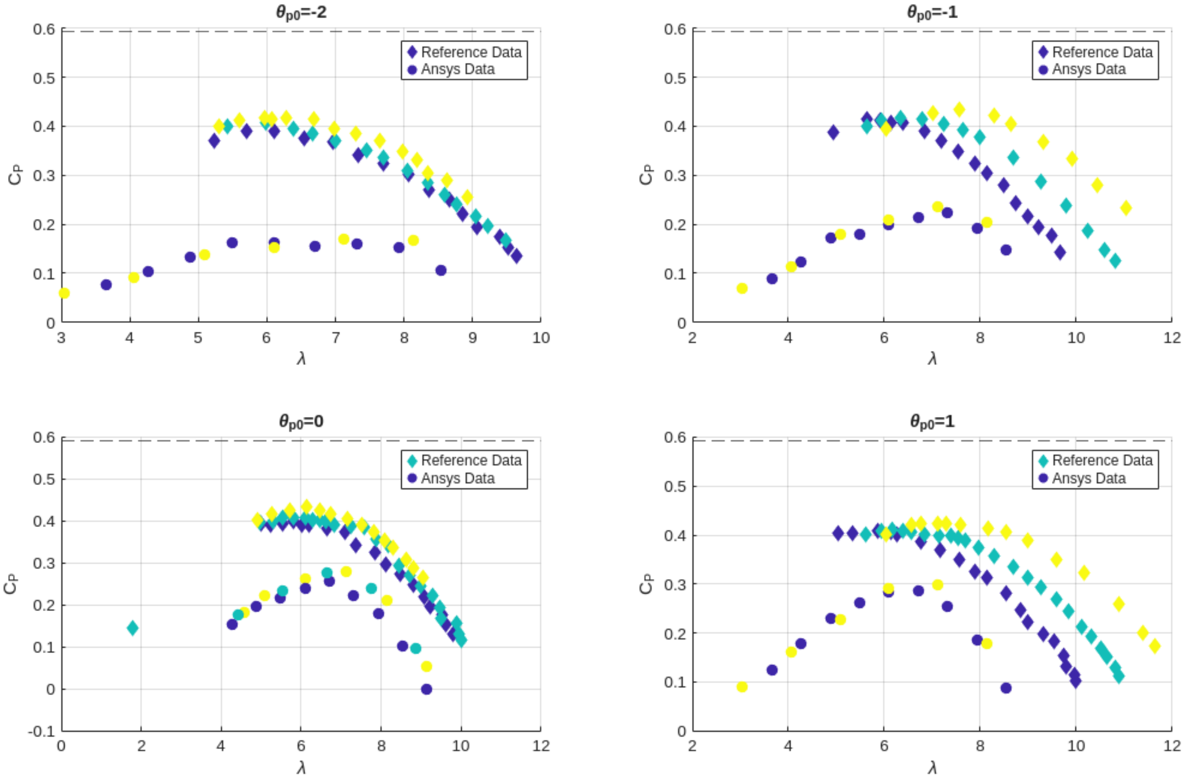


Figure 11: Simulation results (circles) plotted against reference data (diamonds). The different colors indicate ambient wind speed, with blue corresponding to 5m/s and yellow to 6m/s. The actual blade pitch angle for the Ansys data in the top-left graph is -1.8 degrees. The dashed line at the top indicates the Betz Limit

performance at constant λ yet different u_0, making it a safe decision to aggregate this data by wind speed. Unfortunately, that is the only good news. Ansys universally underestimates the torque on the blade with the only region of appreciable agreement occurring on the higher end of samples at $\theta_{p0} = 0°$. The overall shape of the distribution is only similar in the most superficial of ways and the tip-speed ratio that maximizes C_P in the simulation data is consistently about 1 unit above that for the reference.

One of the drawbacks of simulation, doubtless exacerbated in a program like Ansys designed to accommodate with near endless user-defined variables and just as many default values to trust, is that it is incredibly difficult to diagnose issues. Case in point, the first set of simulations I performed resulted in torque measurements (particularly at high angular speeds) that corresponded to power coefficients not just greater than the Betz Limit, but in some cases greater than 1. I wound up going back to the start of the process and rerunning the simulations with seemingly no relevant changes, and the result was the data as presented here. Are these strangely low values also the result of a phantom in the software, a single setting deep in the meshing options improperly set? Perhaps it's just a matter of the meshing density being too low or the residual cutoff being too high, or maybe this data is an accurate expression of the limitations of CFD's predictive power. It's impossible to tell without years' worth of experience running these kinds of simulations .

5.3 Matlab Data in Depth

In what can only be described as a vindication of my method, the data presented in Figure (12) shows greater agreement with the reference data than does the Ansys prediction; and in terms of a ratio of accuracy to calculation time (the entirety of the data shown here could be calculated in under 30s) it is a world

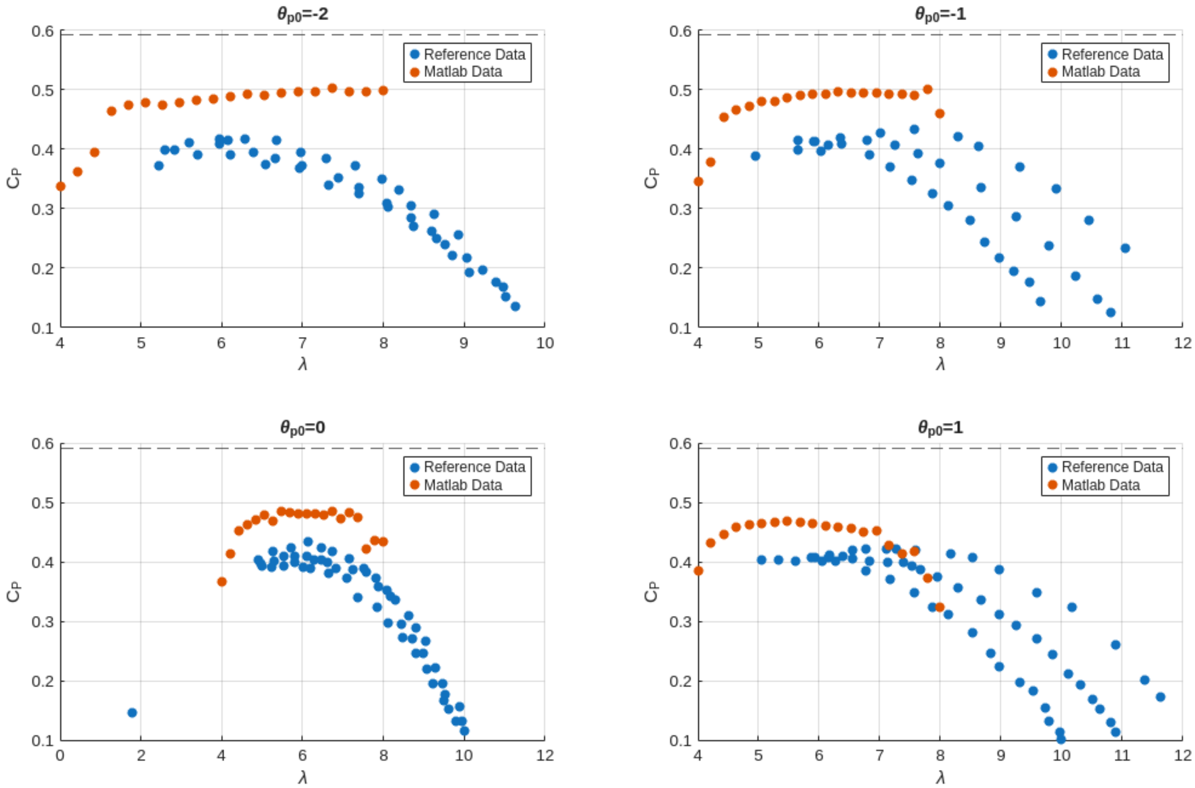


Figure 12: Results of Matlab optimization function on given blade geometry. Velocity information has been aggregated because u_0 does not affect the Matlab optimization. The dashed line at the top of each graph indicates the Betz Limit.

of difference. There is a consistent pattern of overestimating the coefficient of power, but this trend is significantly less pronounced then Ansys' habit of underestimation. The Matlab data also shows greater agreement in the general position and shape of the performance curve, with its estimation of the value of λ that maximizes C_P being far more accurate than the Ansys prediction.

There are a couple important things to note about this data. For one, the blade geometry (pitch angle and chord length) was only provided at 11 points along the blade, which doubtless distorts the value of C_P which is evaluated as a numeric integral of the torque times Ω. In addition, as detailed in Section 4.1, the optimization function returns two different types of error: the first is the proportion of calculated values of angle of attack are outside of the given data range for lift and drag coefficients; the second is the root mean square of the residuals. Another important metric to evaluate this data is the Reynolds number at each segment of the blade, calculated as

$$Re = \frac{c\, u_{eff}}{\nu} \tag{19}$$

with ν being the kinematic viscosity (for air at STP, $\nu = 1.506 \cdot 10^{-5}\, m^2/s$). To convert this distribution of Re into a scalar, the mean is reported. Figure (13) shows how these errors vary with λ at a blade pitch angle of $0°$ and a wind speed of $6\, m/s$. Evidently, the optimizer is far from perfect and suffers from abrupt changes in error magnitude, but these changes don't seem to have much significant affect on the performance metrics. The Reynolds number is linearly proportional to λ, though at a slope that indicates that, given the available options, using the data for $Re = 50000$ is valid for these test conditions.

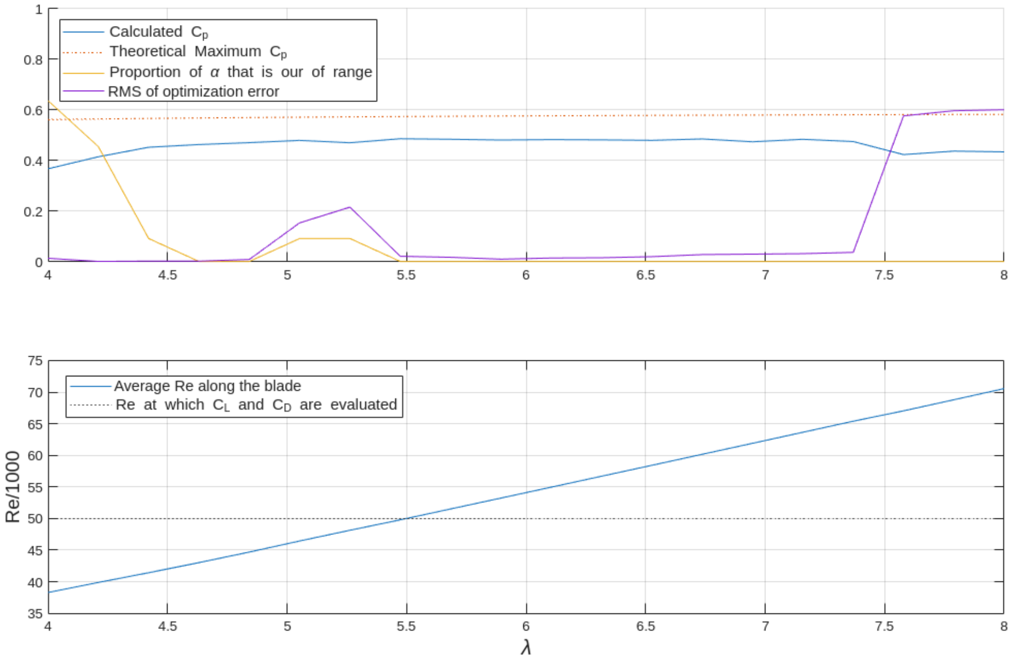


Figure 13: Coefficient of power, average Reynolds number, and optimization errors as a function of tip-speed ratio for the given blade at $\theta_{p0} = 0°$ and $u_0 = 6\,m/s$.

5.4 Performance of New Design

Section 4.2 details the math of how my blade was designed, and its worth noting that the only input parameters to that algorithm are the tip-speed ratio $\lambda = 7$, the number of blades $B = 3$, and the airfoil lift and drag data. Figure (14) directly compares Matlab's predictions of C_P for the given design and my design. Although there are a couple of specific points where the given blade outperforms mine, this is because, while no design value of λ is reported for the given blade, it clearly has a lower operational design range of λ than mine. For this reason, the values of λ at which my blade was evaluated is slightly greater than the range at which the given blade was. When factoring in these different optimal operating conditions, it is clear that my design handily outperforms the competition, reaching a power coefficient between 0.5 and 0.55 over a wide range of operating conditions. Doubtless these values are greater than the true blade performance, but assuming this inflation is of similar degree as that between the reference and Matlab prediction for the given blade, it seems safe to assume that this blade could reach a C_p of 0.5, putting it in competition with real in-use turbines [5].

It should be noted that, though the design algorithm sought to optimize performance for $\lambda = 7$ and returned a default pitch angle $\theta_{p0} = -2.36°$, the testing algorithm returned a slightly higher value of C_P at a slightly greater value of tip-speed ratio and pitch angle. This small difference, and the generally flat nature of the calculated C_P curve, indicate to me both that the assumptions made in the design portion were justified, and that this design has the benefit of having a wide range of operating conditions that result in very-near-optimal power outputs.

Out of curiosity, because of the discrepancy in test values of λ, and because the choice of a design tip-speed ratio at 7 was somewhat arbitrary, I decided to rerun the data collection at a design λ of 4. The results of this perhaps more fair comparison are given in Figure (15). It's clear that the combination of my choice of airfoil and my optimization method produces a blade with greater power production potential, as, with the exception of $\theta_{p0} = -2°$ where they are comparable, the C_P is near-universally greater for my blade than the given blade. Of course, in terms of pure maxima, the blade with a design value of $\lambda = 7$ is still the best option.

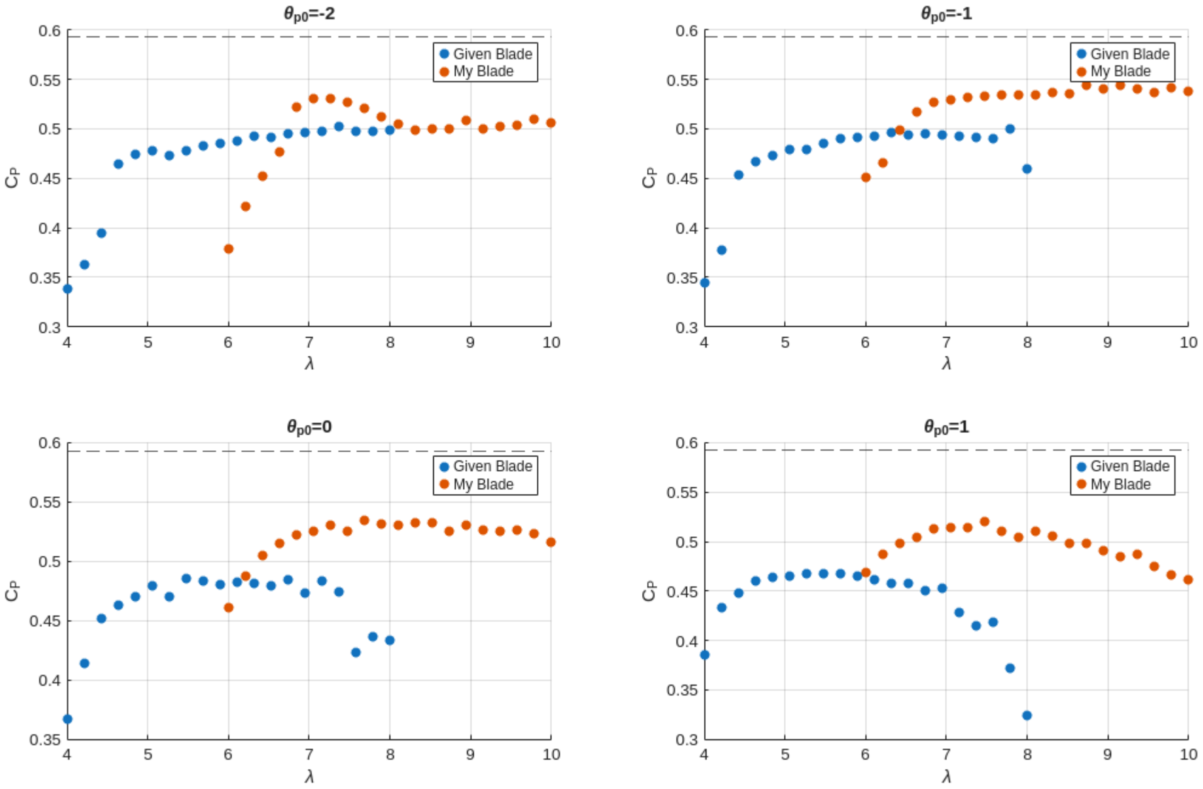


Figure 14: Matlab prediction of coefficient of power as a function of tip-speed ratio at varying blade pitch angles for the given blade design (blue) and my design (red). The data for my blade in the first graph is actually for $\theta_{p0} = -2.3619°$, which is the twist angle generated by the design algorithm.

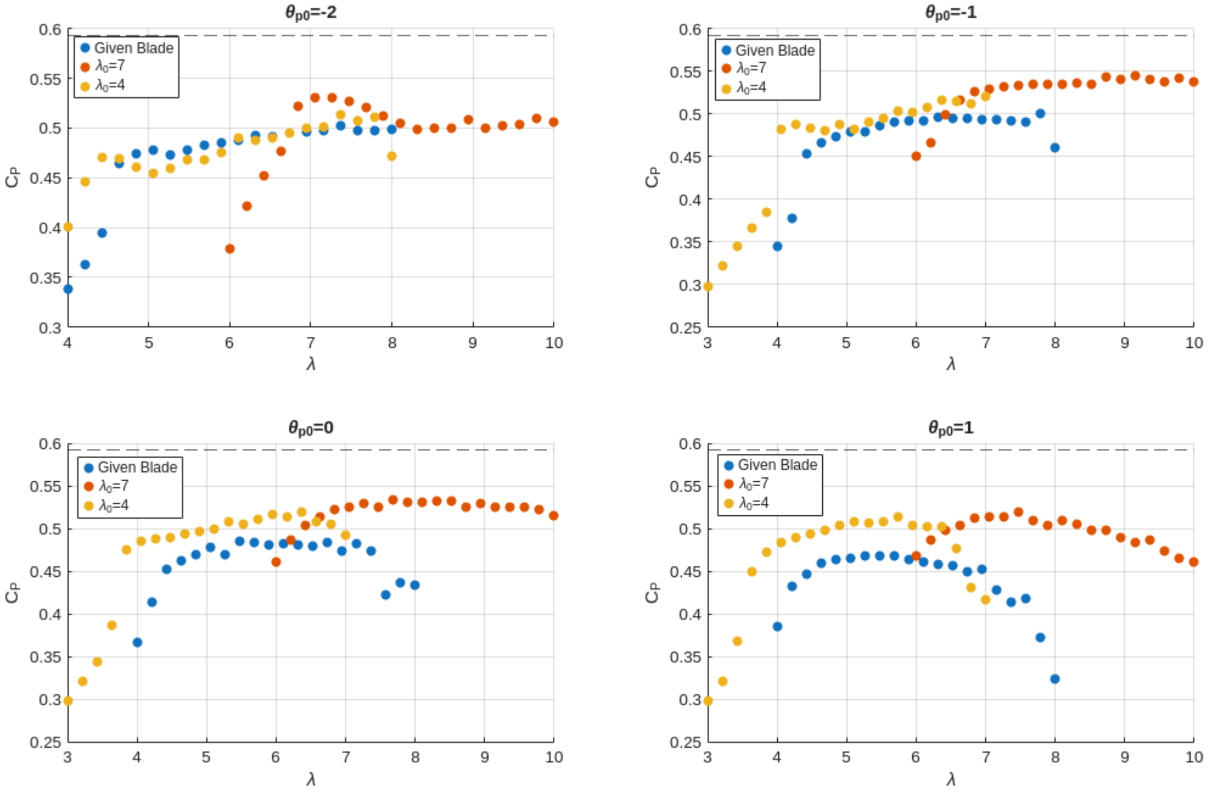


Figure 15: Comparison of turbine performance for the given blade, the design blade, and a modified design using a lower input value of λ.

6 Conclusion

The algorithm designed in this paper far exceeded expectations, in part because CFD proved highly underwhelming as a predictive instrument. It may yet be the case that with the optimal settings and methods, Ansys Fluent could have reported torque productions in great agreement with the experimental data, but the time investment required to implement and test each change is enormous and quickly requires becoming an expert in the intricacies of the software. Meanwhile, in the time required to collect a single data point, the Matlab file could be run dozens of times, each providing (with apparently superior predictive power) data over a whole range of operating conditions. True, the residuals on Matlab's optimization were higher, but I suspect this is primarily a result of data availability; the discontinuous sampling of lift and drag coefficients and their absence at extreme angles of attack ensured that there was no exact solution to the equations `fsolve` was given.

I am surprised that so much of my procedure proved to be nonstandard, as the departure of my math from the equations given in [4] are actually quite small, yet they allow for a significantly simpler objective minimization process than that detailed by the instructor. Unfortunately, the key insight of my derivation, the introduction of two separate drag factors F_1 and F_2, did not really manifest in the data analysis portion of this report. F_1 would be relevant to the calculation of the bending moment on the blade, but such structural considerations were omitted from this assignment. Moreover, it is impossible to extrapolate from just the small number of tests calculated on a single blade geometry whether the predictive power of my algorithm would hold in other situations, so clearly a greater amount of data needs to be collected to better evaluate its performance. But these initial results are promising and I hope that this methodology will find use for those engineers working on actual industrial scale wind power projects as a means to more rapidly iterate on their designs.

References

[1] Energy Information Administration. *Levelized Costs of New Generation Resources in the Annual Energy Outlook 2025*. Tech. rep. United States Department of Energy, 2025. URL: `https://www.eia.gov/outlooks/aeo/electricity_generation/pdf/AEO2025_LCOE_report.pdf`.

[2] Christopher Jung, Leon Sander, and Dirk Schindler. "Future global offshore wind energy under climate change and advanced wind turbine technology". In: *Energy Conversion and Management* 321 (2024), p. 119075. ISSN: 0196-8904. DOI: `https://doi.org/10.1016/j.enconman.2024.119075`. URL: `https://www.sciencedirect.com/science/article/pii/S0196890424010161`.

[3] Elena Llorente and Daniele Ragni. "Serrations effect on the aerodynamic performance of wind turbine airfoils". In: *Journal of Physics* (2019).

[4] F. Manwell, J. G. McGowan, and A. L. Rogers. "Aerodynamics of Wind Turbines". In: *Wind Energy Explained*. John Wiley & Sons, Ltd, 2009. Chap. 3, pp. 91–155. ISBN: 9781119994367. DOI: `https://doi.org/10.1002/9781119994367.ch3`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119994367.ch3`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119994367.ch3`.

[5] Briana Niu et al. "Evaluation of alternative power production efficiency metrics for offshore wind turbines and farms". In: *Renewable Energy* 128 (2018), pp. 81–90. ISSN: 0960-1481. DOI: `https://doi.org/10.1016/j.renene.2018.05.050`. URL: `https://www.sciencedirect.com/science/article/pii/S0960148118305718`.

[6] L Prandtl. "Applications of Modern Hydrodynamics to Aeronautics". In: *National Advisory Committee for Aeronautics* (1923). URL: `https://ntrs.nasa.gov/citations/19930091180`.

[7] Marwa Khaleel Rashid et al. "Integrating machine learning and CFD for enhanced trailing edge serration design on a NACA 0012 wind turbine blade". In: *International Journal of Thermofluids* 30 (2025), p. 101446. ISSN: 2666-2027. DOI: `https://doi.org/10.1016/j.ijft.2025.101446`. URL: `https://www.sciencedirect.com/science/article/pii/S2666202725003921`.

[8] Cornell University. *Wind Blade Analysis for Wind Power*. URL: `https://innovationspace.ansys.com/product/wind-blade-analysis-for-wind-power/`.

[9] David C. Wilcox. “Multiscale model for turbulent flows”. In: *AIAA Journal* 26.11 (1988), pp. 1311–1320. DOI: https://doi.org/10.2514/3.10042.

Appendices

A Blade Design Algorithm

Comments have been removed for space considerations, but the original file has been uploaded to Box.

```
function [lr,sig,theta_p,alpha,phi] = bladeDesign(afdata,n,l,B)

lr = l.*linspace(1e-3,1-1e-3,n);

a = @(phi) cos(phi).*(cos(phi)-lr.*sin(phi));
a1= @(phi) sin(phi).*(cos(phi)./lr-sin(phi));
F2= @(phi)(2/pi)*acos(exp(B*(1-l./lr).*csc(phi)/2));

negCp= @(phi) -(8/l^2)*trapz(lr,F2(phi).*a1(phi).*(1-a(phi)).*lr.^3);
phi = fminunc(negCp,atan(1.72./(2*lr+1)));

[Ct,i] = max(afdata(:,2).*sin(phi)-afdata(:,3).*cos(phi));
alpha = afdata(i,1).'*pi/180;

theta_p = phi - alpha;
sig = 4*F2(phi).*cot(phi+atan(lr)).*sin(phi).^2./Ct;
end
```

B Blade Testing Algorithm

Comments have been omitted for space considerations but are visible in the file uploaded to Box

```
function [Cp,Re,flag] = bladeTest(afdata,r,R,B,l,c,theta_p,u0)
Cl = afdata(:,2).';
Cd = afdata(:,3).';

n = length(r);
sig = B*c./(2*pi*r);
lr = l*r/R;
omega = (l*u0/R);

function y = f(phi)
    phi = phi(1:n);
    alpha = phi - theta_p;

    i = arrayfun(@(alph)mini(abs(afdata(:,1)*pi/180-alph)),alpha);

    F2a=(2/pi)*acos(exp(B*(1-l./lr).*csc(phi)/2));
    F2b=(Cl(i)-Cd(i).*cot(phi)).*sig./(4*cot(phi+atan(lr)).*sin(phi));

    y = F2a-F2b + double(phi<0 | phi>atan(1./lr));
end
function i = mini(x)
```

```
[~,i] = min(x);
end

[phi,Y] = fsolve(@f,atan(1.72./(2*lr+1)),optimoptions('fsolve','Display','off'));

alpha = phi - theta_p;
i = arrayfun(@(alph)mini(abs(afdata(:,1)*pi/180-alph)),alpha);
Ct = Cl(i).*sin(phi) - Cd(i).*cos(phi);

flag = [mean(i==1|i==size(afdata,1)); rms(Y)];

u_eff= u0*(lr.*cos(phi)+sin(phi));

rho = 1.225;
nu = 15.06e-6;

dQ = pi.*Ct.*sig.*rho.*u_eff.^2.*r.^2;

P_wind = 0.5*rho*u0^3*pi*R^2;
Cp = trapz(r,omega*dQ)/P_wind;

Re = mean(c.*u_eff/nu);
end
```