

Modeling the Twisted Savonius Wind Turbine Geometrically and Simplifying its Construction

Nick Halsey

Oregon Episcopal School

February 23, 2011

List of Contents

Abstract.....	2
Introduction.....	3
Methods.....	14
Results.....	22
Discussion.....	28
List of Citations.....	29
Acknowledgements.....	30
Appendices (A-E).....	31

Abstract:

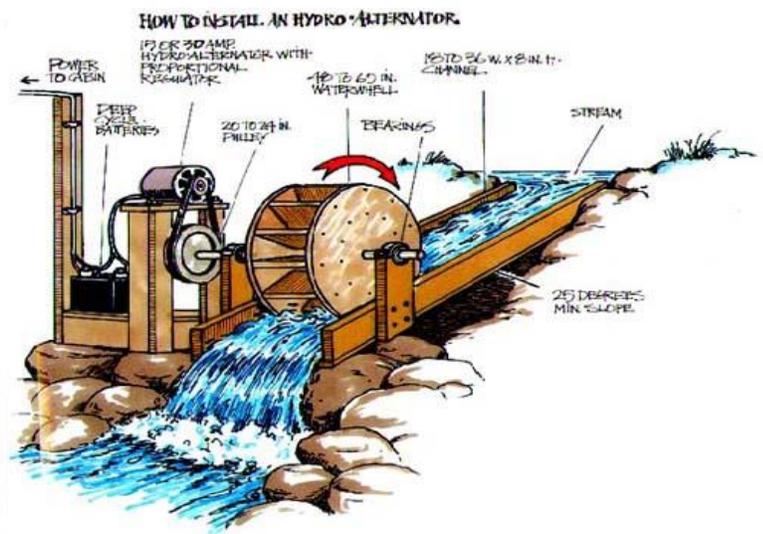
The drag-based Twisted Savonius Vertical Axis Wind Turbine (VAWT) has shown promising applications for use on the tops of buildings, enabling clean energy production at the site of its use, virtually eliminating transportation losses. Unfortunately, the turbine's shape is very complex and three-dimensional because of its twist, requiring complex machinery to construct. I was able to model the geometry of the shape with the symbolic geometry program Geometry Expressions, developing visual models that depict the appearance of the turbine in operation and show the effects of twisting the blade. Ellipses, loci and traces comprised the visual model. The most significant finding was that the radius of the turbine is squeezed as the turbine is twisted, which occurs because of the geometric principles of the blade, not just the limitations of the materials. A greater angle of twist results in a greater potential efficiency in operation. Utilizing the calculus principles of definite integrals allowed creation of an approximation of the shape, "unrolled" into a flat surface using triangles. This can be used to build the turbine much more simply and, with refinement, could allow widespread use of Twisted Savonius turbines on rooftops with little cost relative to other alternative energy options.

Introduction

According to the Energy Information Association, the average price of electricity for commercial consumers increased from 7.26 to 10.28 cents per kilowatt-hour from 1999 to 2008, rising 42% (Energy Information Association, 2009). Government policies also provide monetary incentives for generating clean energy. For the commercial consumer, this creates an incentive to explore alternative energy options such as wind, solar, and hydro power independently to reduce or even eliminate dependence on the electrical grid for power.

Unfortunately, hydro power is not a good option, or even possible, for a commercial building independently producing energy unless there is a small river nearby. In that case a waterwheel could provide some power to the building if there is strong enough water flow (see figure 1). Hydro power could also be utilized in gravity-based rainwater runoff applications, which could be used where there is no access to a creek or river.

Figure 1: A waterwheel that could be installed in a small river or creek (Hydroelectric Power, (n.d.)).



Solar panels are easily mounted on roofs or the sides of buildings; therefore, they are an easy way to generate power for buildings. The disadvantages of mounting panels on roofs are the initial cost and that they will never produce electricity during the night; so enough power must be



Figure 2: Solar panels on a roof (rooftopsolar2, 2010)

generated and placed into batteries during the day to last through the night. The other major problem is exposure to the sun, which varies throughout the day and is obstructed by

other buildings in large cities, as well as varying at different latitudes and times of year (Facts—about-solar-energy.com, 2006).

The final major option is to mount wind turbines on the top or side of buildings. This is difficult to effectively accomplish because wind patterns are very turbulent around buildings. Additionally, most traditional horizontal-axis-style wind turbines vibrate when operating which can be a threat to the structure of the building, the turbine itself, and the people inside the buildings. Wind turbines can work on buildings if they are constructed with a vertical axis design.

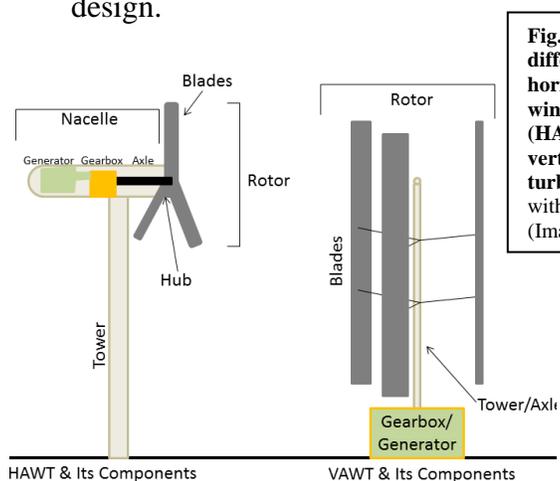


Fig. 3: General differences between horizontal axis wind turbines (HAWTs) and vertical axis wind turbines (VAWTs) with parts labeled. (Image by Author)

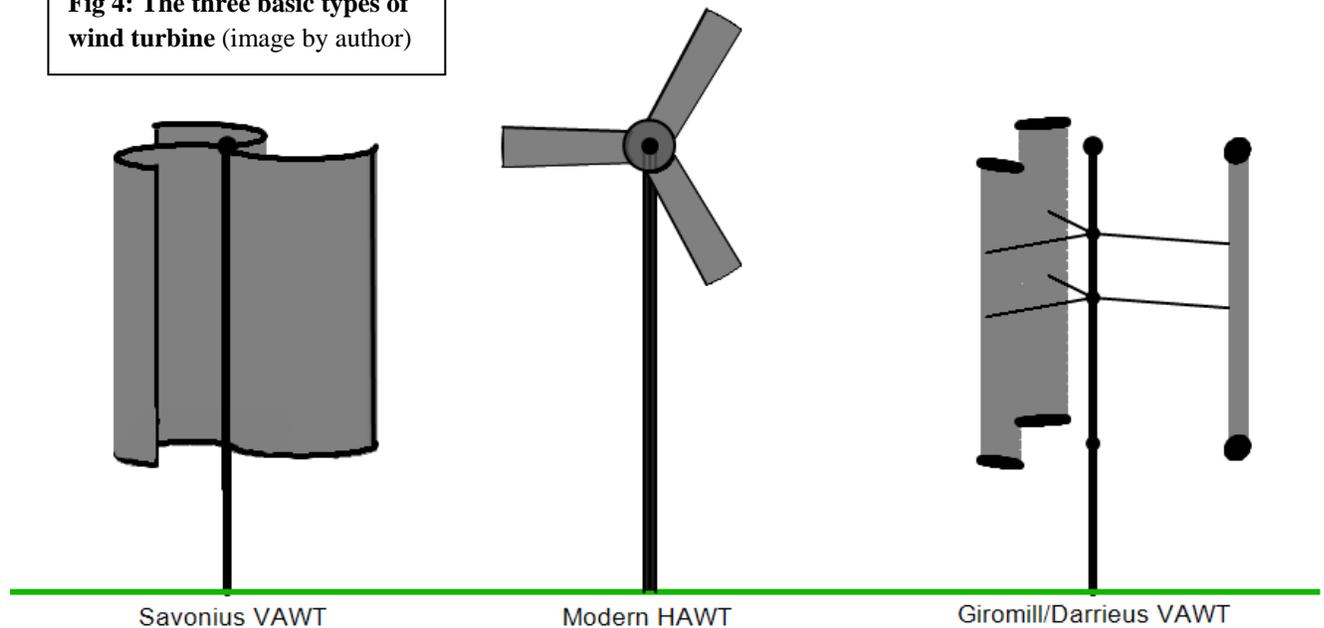
A horizontal axis wind turbine (HAWT) has a vertical tower with a horizontal axis. The blades are straight and attached to the

end of the axle. On a HAWT, wind hits the blades, usually turned at an angle, and pushes them out of the way. Because the blades are angled, this causes the wind

to rotate the hub. The blades can be either flat or almost flat on one side and more significantly curved on the other side, similar to an airplane wing. The number of blades is different depending on the intended usage of the turbine; but if it is for electrical generation, the three-bladed design is most common.

A vertical axis wind turbine (VAWT) has a vertical axle which also acts as the tower. There are many different types of VAWTs, but they all have the gears, generator, and electronics at the base of the tower which eliminates the need to have wires going to the top. The major advantage is that the wind can come from any direction and the turbine doesn't need to be pointed into it.

Fig 4: The three basic types of wind turbine (image by author)



HAWTs are very different than their VAWT counterparts. On HAWTs, the number, size, shape, material, and pitch of the blades can vary, along with many other similar parts within the tower and nacelle (see figure 3); but they commonly have three spinning blades on a horizontal axis perpendicular to a tower. In contrast, VAWTs come in two very different types of design: Savonius and Darrieus (Vertical Axis Wind Turbine, 2009). Figure 4 shows these different wind turbine designs.

The two designs are complete opposites in fundamental blade operation principle, but identical in basic operation and mechanics. The basic Savonius design is drag based—the wind essentially pushes on the blades. The Darrieus design, on the other hand is lift based—the wind essentially pulls the blades because of pressure differentials. Additionally, both of these designs can be twisted about the axle, giving many advantages to the turbine’s operation but being much more complicated to construct.

Most Darrieus VAWTs have thin blades shaped like eggbeaters, but others use straight blades oriented vertically away from the axle. Darrieus blades are shaped like airplane wings

(see fig. 5); a lifting force is found on the leeward side of the blade when the wind hits it, causing it to turn. Some Darrieus turbines require their generator/alternator to act as a motor for the turbine to start operating, while the wind continues this initial motion to generate power. Another disadvantage of the Darrieus design is that the torque created when the wind hits each blade causes pulsations that reduce the life of the turbine and create inefficiencies in operation.

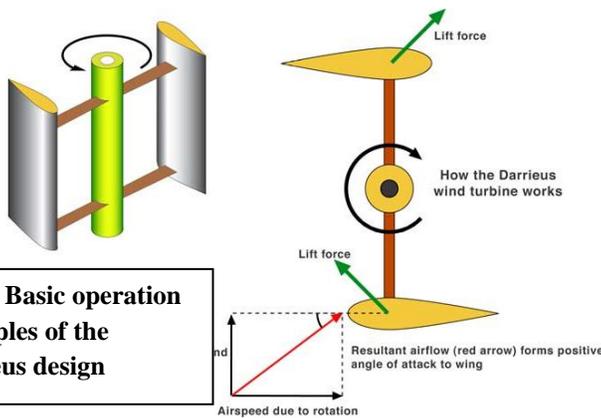


Fig. 5: Basic operation principles of the Darrieus design

Some designs use helical twists, distributing these pulsations over the 360 degrees (ex. 3 blades (offset twisted 60°) of a single rotation (Darrieus wind turbine, 2009 [including fig. 5]).

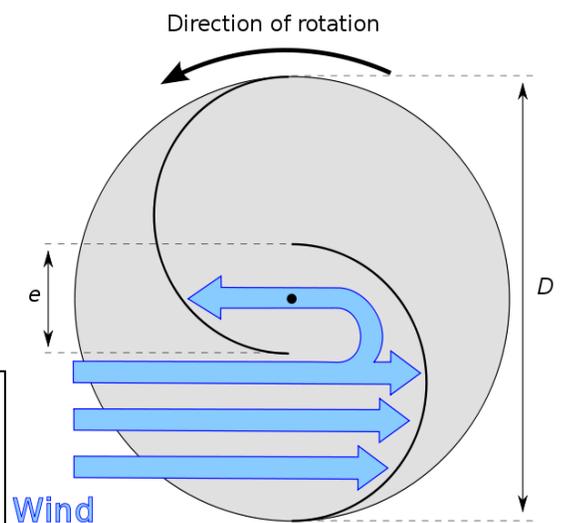
Savonius turbines are drag-based and generally not as efficient as Darrieus turbines.

Savonius designs operate similarly to cup anemometers¹ which are also drag-based. Less drag occurs when the wind is pushing against the scoops than when the scoops are moving with it; this differential causes the torque and rotation on the shaft. Despite only having one relatively

simple design, the shape, the size and number of the blades, the distance “e” (on fig. 6b) can vary on Savonius turbines, and multiple turbines can also be stacked on top of one another (see figure 6a) (Savonius wind turbine, 2009).



Fig. 6a&b: Left: A Stacked Savonius turbine, with three blades per sub-turbine. Right: Diagram of wind patterns within a two bladed Savonius turbine. (Savonius wind turbine, 2009)



¹ A cup anemometer is a device that measures wind speed using several cups that are blown by the wind, spinning around a pole/axle that is vertical.

Despite being relatively inexpensive and requiring little maintenance, Savonius turbines aren't often used because they are inefficient. They will also only spin as fast as the wind is blowing because they are drag based; thus, their tip speed ratio is less than or equal to 1. Savonius turbines, therefore, have slow rotations per minute (RPM) ratios and high torque creating the need to be geared up² when used for power generation. This requires more RPM to provide more wattage (unit of measurement of electrical power output). Fortunately, Savonius turbines capitalize on turbulent winds like those found near buildings, creating higher efficiency on buildings. Because of these factors, Savonius designs are rarely used industrially but are often successful when independently constructed to provide power to buildings (Savonius wind turbines—Wind, 2008).

All of these VAWT designs are better suited than HAWTs for rooftop mounting for several reasons. First, they don't require a tower (apart from the tower that also serves as the axle). Additionally, because a tower is needed for a HAWT regardless of placement, it might as well be placed next to the building on a taller tower. HAWTs also must be kept pointing either into or with the wind to operate, unlike VAWTs, and VAWTs don't experience the damaging vibrations that HAWTs do.

Vibrations caused by wind turbines have harmful effects on both the turbines themselves (including any structure they're mounted on) and the people living or working near the turbines. Although this statement is true for all wind turbines, it mostly refers to HAWTs. It is common knowledge within the renewable energy community that large wind farms, sites with hundreds of turbines up to 400 ft. tall, create challenges for property owners. In addition to the "eyesores" and resultant declining property values are the concerns of noise and health issues. Both the noise and the health problems are caused by vibrations created because of turbulence that occurs as the HAWT's blades spin and pass the tower. The turbulence of the wind is caused by the blades spinning in the opposite direction of the wind (moving perpendicularly to it), and further variation in airflow causes additional vibration each time the blades pass the tower. The vibrations occur at a low enough frequency that they are inaudible to humans, but the

² To gear up the rotation of an axle, a large gear is placed on the first axle and a small gear is placed on the second axle; the cogs lining up cause the second axle to spin much faster.

vibrations are often amplified when multiple turbines are near each other and are spinning at the same frequency, which is often the situation with wind farms. The low frequency vibrations can also cause certain parts of buildings, such as windows, to vibrate. These can cause noticeable noise and potentially damaging vibrations to buildings and people (Van Den Berg, 2004).

Turbine tower height is also a potential issue when turbines aren't mounted on buildings. The American Wind Energy Association (AWEA) continues to attempt to justify placing turbines on the tallest posts possible. Some of their reasoning is logical, such as that faster winds are found at higher elevations. Others, such as a seemingly vital need to avoid turbulence, are not necessary. This opinion is quite clear in the following passage from an AWEA guide designed for permit lawmaking reference by state and local governments:

Why Do They Need To Be Tall?

A tall tower is the single most important factor in the economic viability of a small wind system. Tall towers enable turbines to access faster and better quality winds, and even small increases in wind speed translate to exponentially more energy the turbine can generate. In other words, a taller tower means far more - and cheaper - energy. (AWEA, 8)

Additionally, the AWEA generally disapproves of turbines mounted directly on rooftops, insisting that a tower should be utilized on top of the roof to bring the turbine above all of the turbulence of an urban environment. The issue is, why not save on cost and potential dangers of a post (proposed by the AWEA to be 80'+ long) by designing a turbine to capture those turbulent winds? This excerpt shows the AWEA's favor for tall towers:

Rooftop Turbines and Urban Environments:

Site-ing [turbine placement] becomes especially important for turbines in [rooftop] settings. Wind patterns behave very differently around buildings and in densely-built areas, so a turbine must be sited very precisely in order to gain access to wind of sufficient quality. Height, for example, becomes increasingly important in order for the turbine to rise above aerodynamic obstacles and turbulence... (AWEA, 16)

But what if an 80 foot tower isn't an option? Can a turbine be effective on a rooftop or nearer ground? Most importantly can wind turbines be incorporated into existing structures? The AWEA passively offers a suggestion through display of the picture, right, in the margin of



the brochure (with no caption or mention). If turbines are built into existing towers, like lampposts or telephone poles, then using a tower may be worthwhile. Otherwise, it makes more sense to utilize turbulence-tolerant designs on rooftops, which actually create their own windflow patterns that can channel the wind into the turbine if placed correctly.

Important to remember when considering all of this criticism is that it is not only about HAWTs, which I have already determined unfit for rooftops, but also with large wind farms which magnify the vibrations greatly by combining large turbines and large numbers of turbines. One source explains that the vibrations of a small HAWT can be channeled into the ground through the tower of the turbine. It also warns that if mounted on the side of a house the vibrations will be channeled into the house and that, even with rubber mounts, only 2/3 of the vibrations will be stopped (Wind turbines noise and vibration, 2009). If turbine vibrations go through the house or building, they can have even worse effects than those of wind farms. Therefore, a HAWT placed on a building must have a method of dampening the vibrations it causes so they are not transferred to the building. When a VAWT design is used this issue should be less pronounced because VAWTs don't carry the same vibration problems as HAWTs (although certain designs have pulsations as mentioned previously).

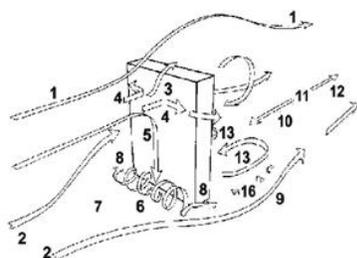


Fig. 7: Wind flow patterns around a building (studied to examine pedestrian discomfort) (Blocken, (n.d.))

Wind flow patterns around buildings are studied for many reasons including snow drifts, dispersion of pollutants, pedestrian discomfort at the ground level, and, of course, wind turbine placement. This diagram shows the flow of wind around a building as determined through studying pedestrian discomfort around buildings in windy areas. There are several places where circular eddy currents form. The wind just goes around the building, generally; but in these places the wind is

very turbulent and, therefore, hard to capture. The ideal placement for a wind turbine on a roof would be at the front edge, closest to the direction from which the wind is coming; however, the wind usually doesn't come from a single direction. The center of a roof would be ideal for any wind direction; however, the turbine would need to be high enough to avoid the eddy current that forms in the center of

buildings with a greater depth than the one in the fig. 7³. The worst place is probably on the edge of the roof opposite from where the wind came, because the strong wind is higher and the eddy currents are gone. Because of all these factors the best turbine placement is high in the center of the roof or near the edge where the wind will most often come from. The turbine should still be designed to capture turbulent wind, though, because of the general nature of building wind patterns.

In qualitative studies conducted on a scale model of a real building, the south end of Oregon Episcopal School's Middle School, fans were used to simulate a randomized wind flow pattern around the building. The wind was designed to be as turbulent as possible to simulate the flow of real winds around a building. I found that the wind hitting the flat side of a building created a concentrated channel of wind ideal for turbines placed directly on the leading edge of the building. When the wind hit at one of the building's corners, however, the flow created a "dead" spot where there was almost no airflow at the leading corner of the roof, but there was a very strong flow along the edges of the roof further away from the corner. Therefore if there is one prominent wind direction in an area, then the wind turbine(s) should be placed along the edge of the roof leading into the wind for maximum power potential (Halsey, 2010).

I have also done quantitative analysis of the best design for VAWTs on rooftops using scale model testing. I found that an H-Darrieus VAWT was more effective than a simple Savonius VAWT, but both designs were inconsistent in operating in the turbulent winds and couldn't always self-start. The Twisted Savonius design, on the other hand, was very consistent in operation and also had much higher average power output, measured in voltage and resistance, then calculated to power output in watts. Therefore, it was concluded that the best type of wind turbine for the turbulent winds found with rooftop use is the Twisted Savonius. The problem is that the process of constructing a Twisted Savonius wind turbine is currently extremely complex, requiring expensive materials and machinery to build and therefore making the cost very high. This is because the shape of the Twisted Savonius design is very

³ The building is more cubic, the top is squarer in shape.

complex. In order to simplify the method of this design's construction, its shape must be better understood (Halsey, 2010).

A study by Saha and Rajkumar (2005) investigated different twist angles for a three bladed Savonius VAWT through low speed wind tunnel testing. They state that previous testing had investigated many of the previously discussed variations of the Savonius design. All of these factors were held constant between all of the models, with the only variation being in the angle (alpha) of twist. Even the model weight was kept constant, at 126.5g. This was accomplished by holding the top of the blades in the same position and radius as the untwisted blades and bringing the bottoms of the blades into a smaller radius than at the top. Each model was about 220 mm in height, and was tested with wind sent through a wind tunnel to create a stable flow. Forces on the models were observed, including torque and angular

velocity (rpm), the primary dependent variable. The graph at right shows the velocity of each of the models tested at various wind speeds. This type of graph represents the "power curve" of each of the designs, when lines connect the points. It was also found that the twisted blades had a much more constant torque than the straight ones, which means that they will require less maintenance and have a longer life-span. General conclusions from the study show that there are advantages and disadvantages to

increased twist angle, but the performance is ultimately improved as the blades are twisted more. The Savonius design was previously known to self-start well, but adding the twist facilitated this trait and created a smoother operation and a generally more efficient design, according to the study.

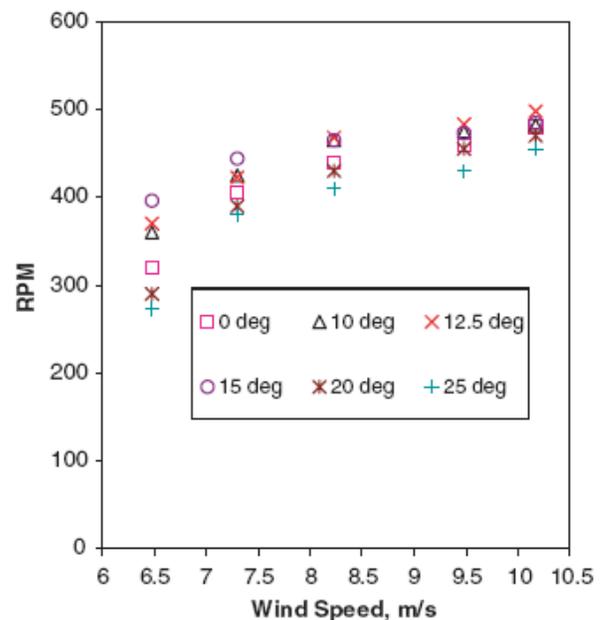
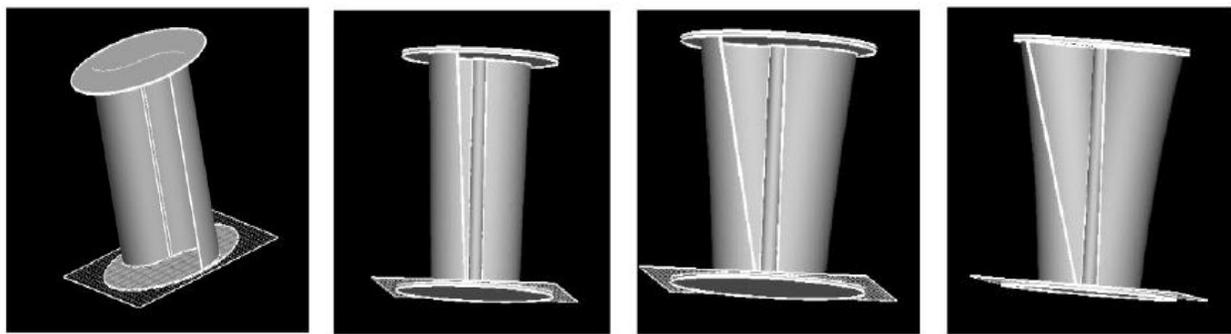


Fig 8: RPM versus wind speed at different twist angles (Saha, 2005)

Another study on the Twisted Savonius design by Hussain, Mehdi, and Reddy (2006), attempted to increase the efficiency of Savonius VAWTs to increase the opportunity for their currently limited use, by twisting the blades. Unlike the previously described experiment, this one investigated twist angles from 0° to 60°. They used CFD which is a program called computational fluid design, in other words, the turbines were modeled and evaluated through a computerized simulation program. Designs were created for twist angles at intervals of 5° from 0 through 60 (see below). After configuring the software to place a simulated wind-flow on the designs, the simulation was carried out and performance data was collected. One of the many output



0 degree twist

10 degree twist

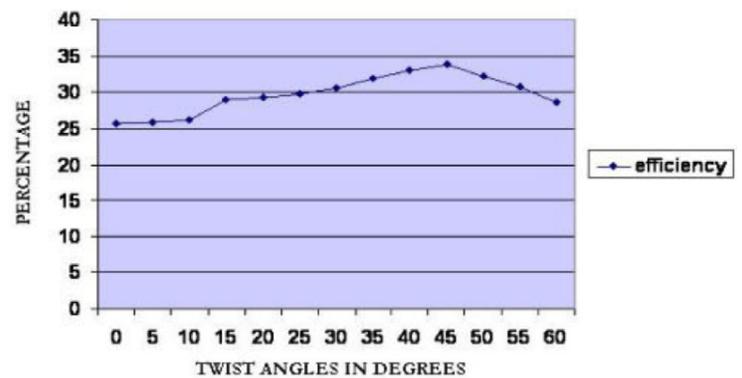
30 degree twist

50 degree twist

values was efficiency. In the graph below, it is clear that increasing the twist increases the efficiency of the model, until the twist reaches 45° at which point efficiency begins to decline. In conjunction with the maximum efficiency at 45° of twist it was found that there was also the greatest positive surface area when the twist equaled 45°. The surface area was calculated by the concave area of one blade minus the convex area of the other blade, so that the forces would produce a net force on the concave blade, spinning the turbine.

Fig 9 & 10: Twisted Savonius Models and their efficiency in terms of twist angle (Hussain, 2006)

Twist Angle Vs Torque



From these sources it is clear that increasing the twist angle of a Savonius wind turbine's blades increases its efficiency, at least to a point, when models are tested numerically in terms of velocity and efficiency. Theoretically, the more twisted a turbine is, the more efficient it will be until the angle of twist reaches 180° , at which point the blades of the turbine reach all the way around the axle of the turbine. So why does the efficiency decline from 45° on? In order to understand this crucial piece of the twisted Savonius design—the twist—and then determine a better method of building it, it is necessary to get to the root of the blades' shape. Scale model testing and computerized airflow modeling have been done to explore the different aspects of optimizing the twisted Savonius design, but it is necessary to explore the geometry of the blade shape to understand why there is an ideal value of twist. Exploring this value, thought to be at 45° , could provide crucial information about the geometry of the shape that suggests why it works so well fluid-dynamically. It could also explain why more twist reduces efficiency after 45° of twist. How should the design be modeled and explored, though? Since the goal is to model the design geometrically and be able to easily adjust the parameters of its geometric structure, the use of a symbolic geometry program is ideal.

Symbolic geometry programs can draw accurate geometric drawings and use numbers, variables and/or equations to constrain the drawings. These constraints can involve many variables which can be adjusted or even animated through a tool within the program. In addition to creating these extremely accurate and mathematically correct drawings, symbolic geometry programs can make calculations of lengths, angles, and more within a drawing, both numerically and symbolically in terms of the variables used (Oke, 2010 [both sources]). This project utilizes the most prominent symbolic geometry program, *Geometry Expressions*. Although its drawing plane is two-dimensional and the complex turbine shape is very three-dimensional, ellipses can be used to represent circles that have been tilted into a three dimensional angular view and cross sections of the turbine can be viewed. Therefore, the two-dimensional software is still very useful for modeling the twisted Savonius turbine.

The objective of this project is to explore the geometric shape of the twisted Savonius VAWT in order to optimize the design and develop a simpler method of constructing it. In investigating the shape, a particular focus was on the differences in shape as the angle of twist is varied. I predicted that there is something causing a reduction of surface area within the geometry of the blade that causes it to lose efficiency at higher levels of twist. This was all done with a geometric model in the program *Geometry Expressions*, allowing observations to be made on both still and animated images. Calculations on different pieces of the model were also used to learn more about the shape including an attempt to find the equation for the shape's surface area in terms of the turbine's radius, the twist angle and other parameters (which involves advanced calculus to determine). It may also be possible to determine the three-dimensional parametric equation for the surface of the blade. After the geometry of the blade was studied, methods for "unrolling" the blade into a flat shape that could be stretched over a frame to build the turbine were studied, again using *Geometry Expressions*. If successful, this method would drastically reduce the cost of the turbines. By reducing cost, widespread use would become much more feasible and generation of wind energy at the location of power usage could become a regular practice.

Methods

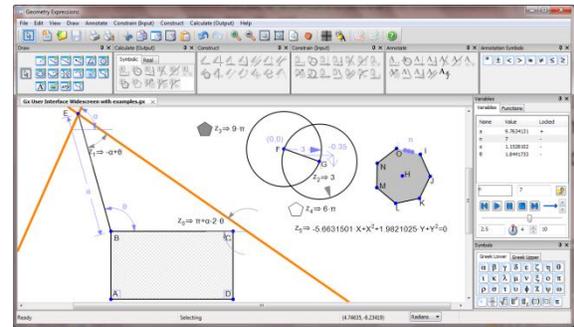
Procedures:

Software Background: The two software programs used for this research were the symbolic geometry program *Geometry Expressions* and the computer algebra system *Maple*.

Geometry Expressions was originally published in 2006 and is used primarily as an educational tool for high school and college levels of mathematics classes (Algebra and Geometry through advanced Calculus). Several currently yet-to-be-published versions of the software (2.3.00-2.3.07, 2.3.21) were used along with the newest released version, 2.2; however, the only unreleased feature used is an animation exporter which is not necessary for making the model, but is for using it. A full list of features is in Appendix A.

Maple is a comprehensive computer algebra system made by *Maplesoft*. It was not used as heavily as *Geometry Expressions*, but key features are algebraic and calculus manipulation of mathematical expressions and 2 and 3 dimensional plots. The operation is similar to that of an advanced graphing calculator (such as the TI-89), but on a computer. The version *Maple 14* was used.

Figure 11: Basic User Interface of Geometry Expressions



Constructing the Models in *Geometry Expressions*:

Both a top view and a side view of the wind turbine were made. The following paragraphs state each step in the mathematical process of constructing them (but not specifically how to do it in *Geometry Expressions*). The first, and easiest, part of modeling the twisted Savonius wind turbine is making a 2 dimensional top view. The twist of the turbine is constrained as theta, the rotation of the turbine is X (which allows us to animate the turbine as if it's spinning in the wind), and s and t represent how far up and around the blade a certain pair of points are. We can “fill in” the entire blade by constructing traces. Traces are essentially the path along which a certain line, curve, locus, etc. moves traced out a given number of times through a certain interval. In other words they are the path of a line, locus or curve as a parameter (usually a point proportional along a curve) changes. These parameters all stay consistent between the different models and the traced surfaces method is also used in the side view model. Note that in this project I assume that the Savonius wind turbine's blades have a semicircular cross section (horizontally). This is often the case with Savonius turbines but some have cross sections that are more elliptical. Here is the entire construction process of the top view:

First, 4 circles are drawn and their radii are constrained to be ‘r’. Next, the intersection point of 2 circles is created and line segments are drawn from the point of intersection to the centers of the circles. The intersection point is set to (0,0) and one of the line segments’ direction is constrained at X. Next, the angle between the line segments is set as theta. Then, 2 points are added to each of the 2 circles being used right now, which will leave 3 (the first is added when the circle is drawn). Arcs are drawn between 2

of the points on each circle. On the first circle, the arc endpoints are constrained to be proportional along the curve at X and $X+\pi$, the third point is $X+t$. The same is done on the other circle, but with θ added to each constraint and the endpoints oriented so the arc faces the opposite direction. This entire process is repeated with the other two circles, and the intersection point from before is coincident with the new intersection so that all four circles intersect in the same place. The angle between the line segments is still θ , but now the angle between one of the new line segments and the corresponding old one is π radians instead of a direction of X . See the screenshot image (figure 12), right.

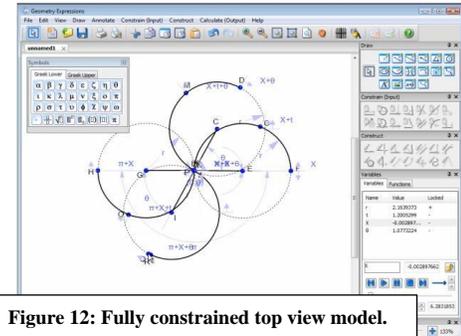
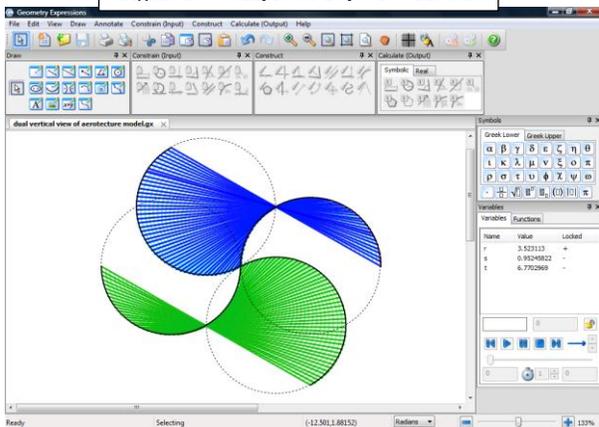


Figure 12: Fully constrained top view model.

Everything but the arcs and the points constrained by t are hidden. Line segments are drawn from one of these points to the next one, rotated at θ . Points are placed on each line and constrained to be proportional at s . The locus of these new points is taken, from 0 to π through parameter t . The trace of each ellipse is taken through the parameter s , from 0 to 1. This will make the blade shape appear. Next, the trace of each of the new line segments is constructed, through t from 0 to π . Now the blade is

Figure 13: Completed Top View Model



completely filled in with a grid. Using different colors can enhance the model additionally. Using the animation exporter feature (unreleased), an animation can be created of the turbine spinning as if in operation or an animation of twisting the turbine can be created by varying θ . An interesting geometric proof from this model is in Appendix B.

The side view model is constructed in pseudo-3D by basing the frame on ellipses which act as tilted circles in *Geometry Expressions*. With this model, the traces give the appearance of an actual twisted Savonius turbine, and when the X variable—which controls the spin—is animated, a video of the wind turbine operating in the wind as it would look from the side is produced. The ability of *Geometry Expressions* to create semi-transparent surfaces out of traces allows both blades to be seen as they are spinning which is especially useful for trying to study and understand the shape.

To build the model in *Geometry Expressions*, three ellipses are drawn first. The equation of each is constrained so that two ellipses are always tangential to each other and the third ellipse and the first two ellipses rotate within the third as X is varied. The following equations are used to constrain the ellipses, with all of the letters and symbols being used as ellipse parameters except for X which controls the general spin.

$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 = 4; \quad \left(\frac{x-h-a*\cos(-X)}{a}\right)^2 + \left(\frac{y-k-b*\sin(-X)}{b}\right)^2 = 1;$$

$$\left(\frac{x-h-a*\cos(\pi-X)}{a}\right)^2 + \left(\frac{y-k-b*\sin(\pi-X)}{b}\right)^2 = 1$$

Next, points are created on the smaller ellipses as they were in the top view and with exactly the same constraints. When all six points have been drawn and constrained, the entire drawing is copied and pasted. The only apparent difference is that the constraints are thicker. Dragging each of the ellipse equations, one of each is revised with a minus ht in the y section like this:

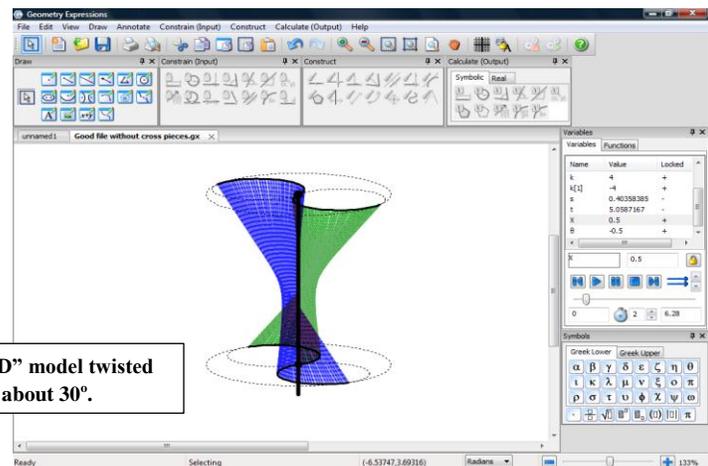
$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-ht-k}{b}\right)^2 = 4$$

When all three ellipses are duplicated below the original ones at a distance of ht, theta is added to every constraint right next to the X (except in the outer ellipse equation) as in these equations:

$$\left(\frac{x-h-a*\cos(\pi-X+\theta)}{a}\right)^2 + \left(\frac{y-k-b*\sin(\pi-X+\theta)}{b}\right)^2 = 1; \quad t + \pi - X + \theta$$

As before, line segments are drawn from the top set's t points to those on the bottom. Then a point is placed on each line segment and constrained to be proportional at s. Finally, loci and traces are constructed as before. The final figure (14, right) can again be animated, in operation or twisting.

Figure 14: The basic side view “3-D” model twisted (theta) about 120° and rotated (X) about 30°.



As will be seen in the results section, geometric

constraints require that multiple sections of blade are necessary in a turbine for it to be twisted the full

180°, which necessitates a slightly extended procedure for construction. Four sections identical to that already described are twisted 45° or pi radians, and stacked to form this more effective design. The properties requiring the stacking process will require it for development of a new method of blade construction also, but fortunately this requires little extra work, only creating four separate sections and stacking them. To create the visual model of the full, stacked turbine, five sets of rotating (with X) ellipses were created instead of two. The variable ht was again used to distance all of the elliptical sections, but this time in fractions so the sections would be spaced evenly. All of the proportionally constrained points on the small ellipses remain, but are all twisted by a fraction of theta (along with the small ellipse equations) so that each layer moves $\theta/4$ radians farther around the turbine as X and theta are varied. Though the construction process is very long and tedious, the side view model can be fully built in about 45 minutes even with all 5 elliptical sections using a method with refined keystrokes. Discovery of this method was also very difficult because of the difficulty in constraining the figure in a tilted, elliptical view. The result, figure 15, right, is worth the effort because it allows one to conceptualize the turbine's shape, especially when it is animated; an easy possibility on a fast computer thanks to the time consuming use of the variables X and theta.

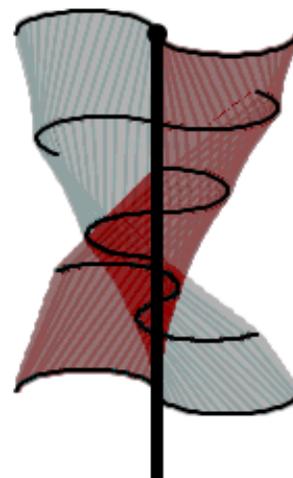


Figure 15: The full side view model

In addition to building and observing a computerized model of the twisted Savonius wind turbine, I also developed a method which allows one to create the turbine from a flat piece of material without specialized equipment. As is the case with a sphere, the shape of the twisted Savonius blade cannot be unrolled into a flat shape (like a cylinder can be) because of the relations in its three-dimensional cross sectional planes. Instead, I used triangles based on control points and line segments taken from the top view model, then incorporated the vertical dimension, a, with the Pythagorean Theorem to create a flat surface with inner fold lines that can be erected into the blade shape by folding in opposite directions. I will briefly explain the process of doing this using 4 sections to approximate, but this method was expanded to include approximations of 6, 8, 10, 12, 14, 16, 18, 20, 24, 28 and 32 sides.

While the full side view model can be constructed in 45 minutes, the higher approximations can take upwards of 3-4 hours to complete, each.

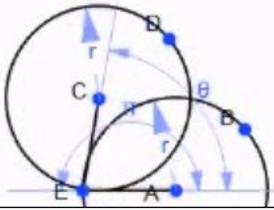


Figure 16: Basic top view framework for building triangle approximations

Figure 16, left, shows the most basic frame for the top view of the turbine, with all of the constraints. The circles represent the top and bottom cross-sections of the model and the variables theta and 'r' are used for the angle of twist and the radius of the circle, respectively; no other variables from the other models were used. Points were constrained around the first half of each circle at even intervals; for the 4 side

approximation this was at 0, pi/4, pi/2 and 3*pi/4. The second (left) circle was constrained identically but with theta added to all of the points. A point was automatically placed at pi for both of the circles because the angle between the circles was theta and the direction of the lower circle radius from which the angle between the circles was theta was constrained to be 0. Next line segments were drawn between the corresponding points (1 on circle 1 to 1 on circle 2, etc.) and on points going between (2 on the first circle to 1 on the second, etc.). These “diagonal” lines between differently constrained points were colored differently in all of the models to reduce confusion (see figure17, right).

Once the circles were completed, the symbolic lengths of every line segment were calculated. Each of these equations represented the adjacent side of a right triangle, which appeared as a line in the top view. The variable 'a' was used to represent the vertical, adjacent side which appeared as a point in the top view. The remaining side of the triangle, the hypotenuse, needed to be found, and then used to constrain line segments in the unrolled shape. Fortunately, the Pythagorean Theorem allows this to be done: the theorem states that in a right triangle with the adjacent and opposite sides being a and b and the hypotenuse being c:

$$a^2 + b^2 = c^2, \text{ so } c = \sqrt{(a^2 + b^2)}.$$

Figure 18: Pythagorean Theorem applied to first line segment

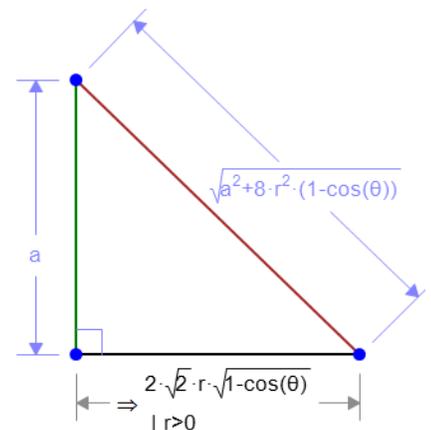
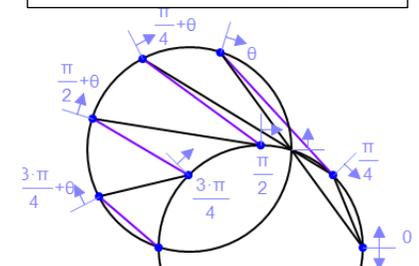
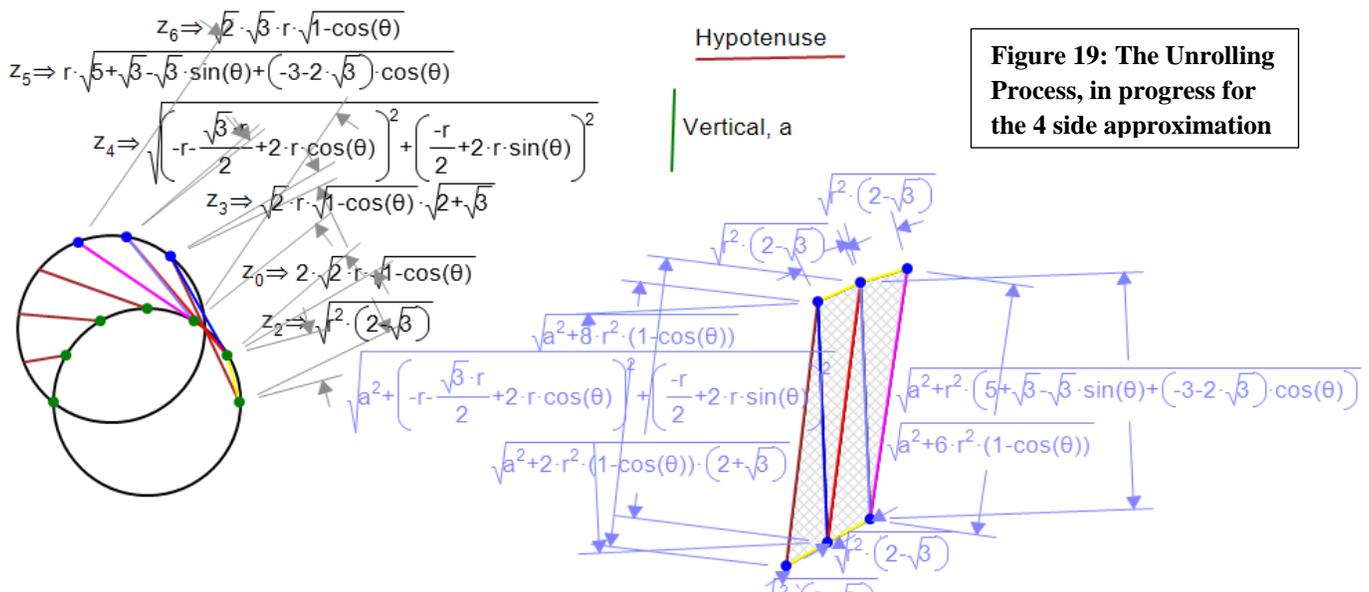


Figure 17: Full frame for line calculations in the top view



Now, because the calculations represent the b side, the calculations must be squared, a^2 must be added and then the square root of the total must be taken to find the constraint, representing the c side. If this process is completed incorrectly for any line segment, the rules of triangle side length constraints will not be fulfilled and the unrolled figure will be reduced to a single line—the one that was improperly constrained. Therefore, it is best to slowly work through the figure and make one calculation, incorporate the variable a and use the resulting expression to constrain the unrolled line segment before moving to the next line segment. One side of each unrolled triangle doesn't need to have a incorporated into it: the 4 sides that sit at the top and bottom of each blade. These constraints are copied directly from the calculation of the symbolic distance between any two consecutive points on the same circle.

All of the expressions are laid out systematically as constraints on the actual unrolled shape. Figure 19 shows the progression of the unrolled figure as more and more of the triangles are added with color coding showing their relations to the circles. The first “vertical” side of the first triangle, left to right, uses the first black line’s equation on the circles: from point 1 to point 1. The short bottom side uses the unchanged distance between two consecutive points on the same circle. The last side of the triangle (also the first side of the next) is the first colored line, from point 2 to point 1. I find it useful to make these line segments colored on the unrolled model as well, because they are folded in the opposite



direction of the black lines. The constraint process continues until all of the lines on the circles have been used, and the last vertical side, representing a point on the top view, is simply a , because the adjacent side is equal to 0. After all of the constraints, point labels, etc. are hidden, the unrolled figure is reflected about the last side, constrained as a , which is also constrained to be oriented vertically. Constraints in the figure can again be animated, but the only logical one for this figure is θ , the twist. Appropriate ranges for all the variables should be input, I used 0 to π for θ and locked a as $1.83/4$ and r as $1/4$, the aspect ratio used by previous researchers (Saha, 2005). At times the aspect ratio was rounded to 2:1 for simplicity purposes. Creating a polygon with all of the outermost side lengths allows for a real calculation of the shape's area. Unfortunately a symbolic output, which would be extremely useful for analyzing shape changes in terms of height, radius and θ changes, does not seem to exist in a finite form, because *Geometry Expressions* cannot complete the calculation after 5 minutes (it usually takes milliseconds) and the partial equation it gives is immensely long. The completed 4 sided approximation is right in figure 20 and the others are found in the results and appendix D.

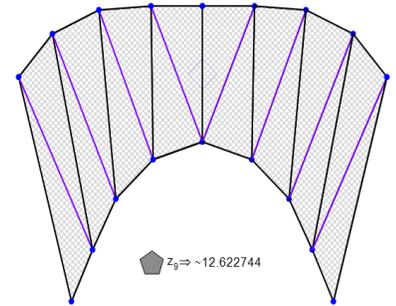


Figure 20: Completed 4 sided approximation.

Observations:

In looking at the figures and animations produced, observations were made about how the turbine's shape functions geometrically. This included mathematical and qualitative investigations of surface area and surface distribution through the blades. Several calculations were made on the figures and then manipulated in *Maple* to produce more graphs and actual three-dimensional images as well as to make algebraic and calculus calculations. Two-dimensional graphs were created in *Geometry Expressions* after manipulation in *Maple* as well.

Data Analysis: No strictly quantitative data were collected, so there was no data analysis in the formal sense. The formulas output from *Geometry Expressions* were often manipulated in *Geometry Expressions* and *Maple* as described in the procedures, which could be considered data analysis. In particular, the two-

dimensional graph that modeled “the squeeze” as the turbine was twisted was used to decide that using 5 cross pieces with a 45 degree rotation between each would be the most effective way of constructing the turbine in a more simple manner. Graphs were created to analyze how close the approximations were to the limit of the surface area for $n=\infty$ triangles (and regressions were calculated on them), utilizing the basic calculus principles of limits and derivatives. In theory, the calculus of integrals can be used to find the exact surface area of the actual blade—not just the triangle approximation—however, this process requires extremely advanced calculus principles. Therefore, the data points for each triangle approximation were manually analyzed and interpreted, much like traditional data collection, so quantitative data analysis was conducted for the surface area investigation.

Results

The Squeeze:

As previously mentioned, the Savonius wind turbine's blade is essentially squeezed as it is twisted without any supports between the twist points, which was discovered when the first animation of twisting the turbine was viewed. To model the squeeze, it was first looked at from the simplest view: the top. The angle between the circles in the top view was constrained as θ and there are proportional points at t and $\theta+t$ on the circles with a line segment between them. By moving the value of s , a proportional point on the line segment, the green locus moves up and down through all of the turbine's horizontal cross sections (fig 21b, below). We can see in fig. 21a, below, that the locus of that point (in green) in terms of t is always a circle but gets larger and smaller:

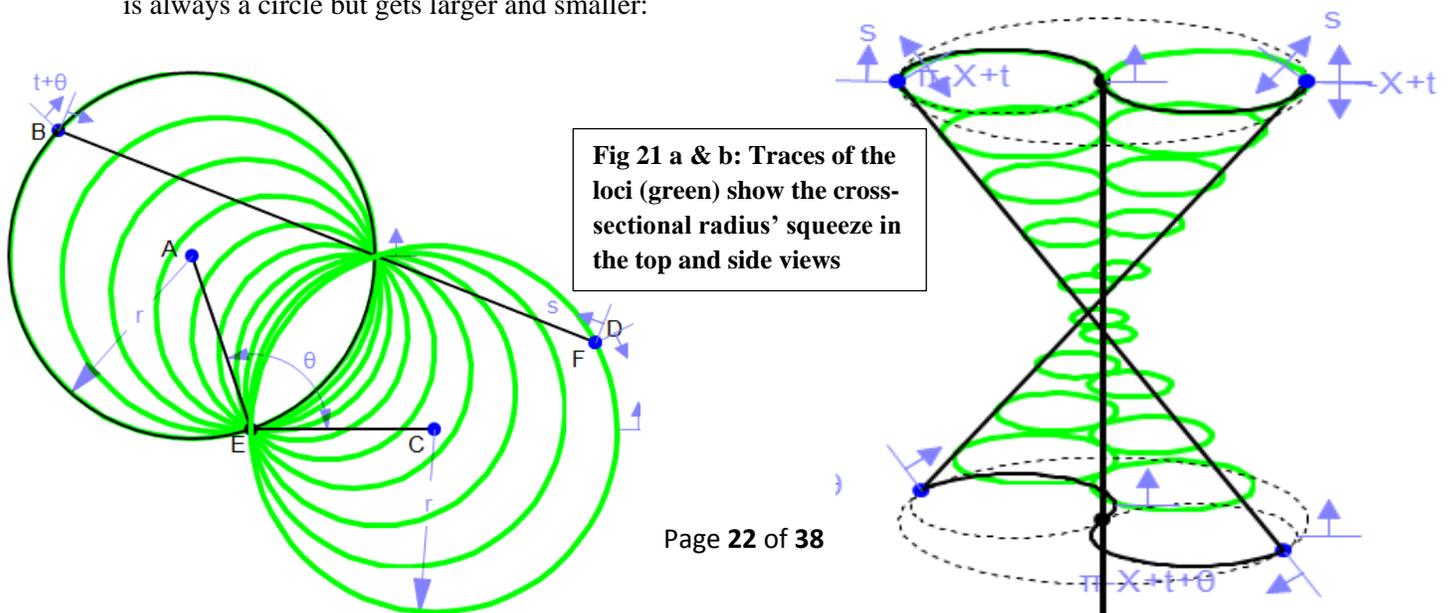
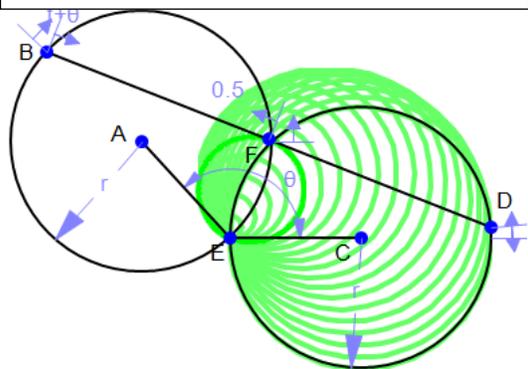


Fig 21 a & b: Traces of the loci (green) show the cross-sectional radius' squeeze in the top and side views

One question that comes out of this squeeze is at what point is the squeeze the most extreme (or where is it most squeezed) in terms of s ? This happens when the distance between the two intersection points of the circle is equal to the diameter of the locus. This is because the locus always goes through two points, the intersections of the circles. How do we know where this is, in terms of s , though? The point s being in the same place as one circle intersection doesn't mean the locus is at its minimum size. It is best to eyeball where s is when the circle is at its minimum size and then replace s with that estimate. It is confirmed by calculating the locus' symbolic equation, drawing a circle and constraining it to have the same equation as the locus, then calculating the circle's radius and the distance between the circle intersections. When the radius is half the distance, the correct value for s has been found. Fortunately, the minimum radius is found at $0.5s$, or halfway between the top and the bottom. This means that the squeeze is symmetrical. Now the question is what effect does theta (the twist) have on the amount of squeeze? Figure 22, below, illustrates how the radius, constrained at the maximum squeeze value

Fig 22: the trace of the locus (green) shows the effect of adjusting theta on the squeeze, when it is at its most extreme point ($s=.5$)



of 0.5 affects the amount of squeeze. Remember, the smaller the locus, the more the squeeze.

The squeeze can also be modeled in a computer algebra system (CAS), such as *Maple*. To do this, the *Geometry Expressions* calculation for the symbolic radius of the locus of the point constrained at s is copied and pasted into *Maple*. Then, the constant r is set at 1 and theta is replaced with π . Plotting the result gives the amount of squeeze moving up/down the turbine when it is twisted π radians (note that *Maple* works almost exclusively in radians). Going back to the theta equation, a three-dimensional plot of the squeeze can also be created. In this plot, the unmarked axis is the radius of the locus and the other two

dimensions represent theta and s. Finally, there is a two-dimensional plot of how the twist affects the radius at the minimum radius and a calculation of the radius at a specific theta value. The relevant pieces of the maple worksheet are below in figures 23 a-l, below.

Fig. 23 a-k: Maple Worksheet

$$\sqrt{\frac{1}{4} (2r - 2sr + 2\cos(\theta)sr)^2 + \sin(\theta)^2 s^2 r^2}$$

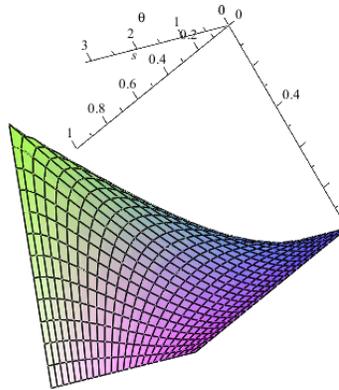
`eval(% r=1);`

$$\sqrt{\frac{1}{4} (2 - 2s + 2\cos(\theta)s)^2 + \sin(\theta)^2 s^2}$$

`eval(% theta = pi);`

$$\frac{1}{4} \sqrt{4} \sqrt{(2 - 4s)^2}$$

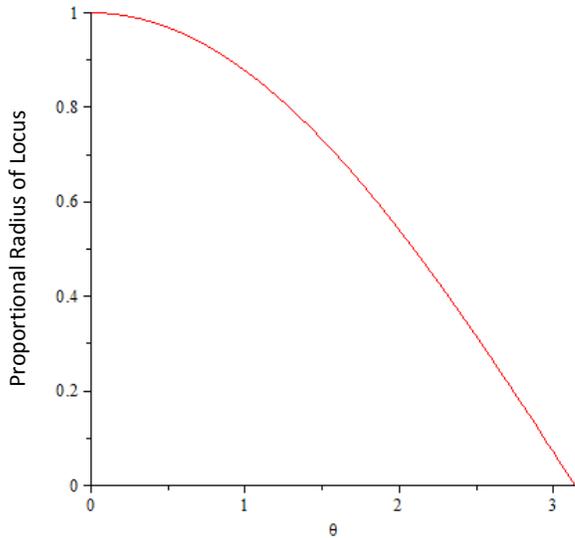
`plot3d(sqrt(1/4 (2 - 2s + 2cos(theta)s)^2 + sin(theta)^2 s^2, s=0..1, theta=0..3.1415927);`



`eval(sqrt(1/4 (2 - 2s + 2cos(theta)s)^2 + sin(theta)^2 s^2, s=.5);`

$$\frac{1}{2} \sqrt{(1.0 + 1.0\cos(\theta))^2 + 1.00\sin(\theta)^2}$$

`plot(% theta=0..pi);`



This last graph (fig. 23k) is most useful because it shows how much compromise in squeeze there is for a given angle of twist on the turbine. Avoiding this issue is the ideal solution, and with the goal being to make the turbine as geometrically simple as possible, it was determined that stacking four sections of turbine, each rotated 45° was the ideal solution for these conditions. This way there is a minimal amount of squeeze ($45^\circ = \pi/4$ radians, $\sim .7$ radians for graphical reference), but also a minimal number of sections to separately construct and stack.

The Surface Area Limit:

As expected, each approximation model had an area that was slightly bigger than the previous, and the difference between each area grew smaller and smaller as the number of sides, n , got bigger. The slope of the line segment connecting the twenty-eight and thirty-two side approximations was only .001349, with the rise being about .0054 and the run being the change in n , 4. Calculating all of the other line segments on the graph, using *Geometry Expressions* as the grapher by constraining points and showing the axes, they generally got smaller and smaller, confirming the visual evidence of the actual graph numerically. The main concepts of calculus, limits, derivatives and integrals are very relevant to this piece. The goal is to find the limit of the surface area as n approaches infinity, which represents a definite integral on an unknown (and possibly non-existent) function. The derivative is used to find the rate of change (slope) between each approximation's data points. Figure 24, below, is the graph in *Geometry Expressions* with the data points and the slopes between each approximation.

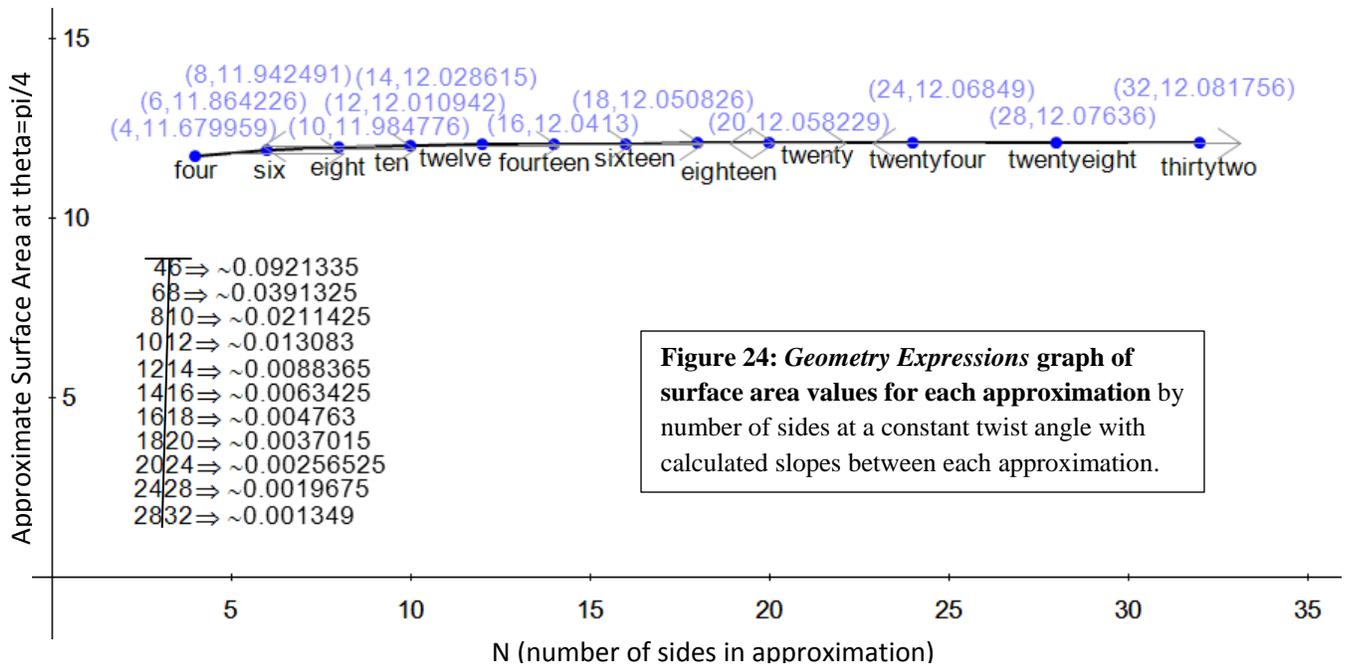


Figure 24: Geometry Expressions graph of surface area values for each approximation by number of sides at a constant twist angle with calculated slopes between each approximation.

In order to show this limit all four ways: geometrically (building models), numerically (areas and rates of change between areas), graphically (with the graph of the area compared to n) and algebraically; it is useful to try to find an approximate equation to represent this limit to give an algebraic representation. Because of the almost logarithmic shape of a finite limit-approaching graph, a logarithmic regression was found to best approximate the graph. The

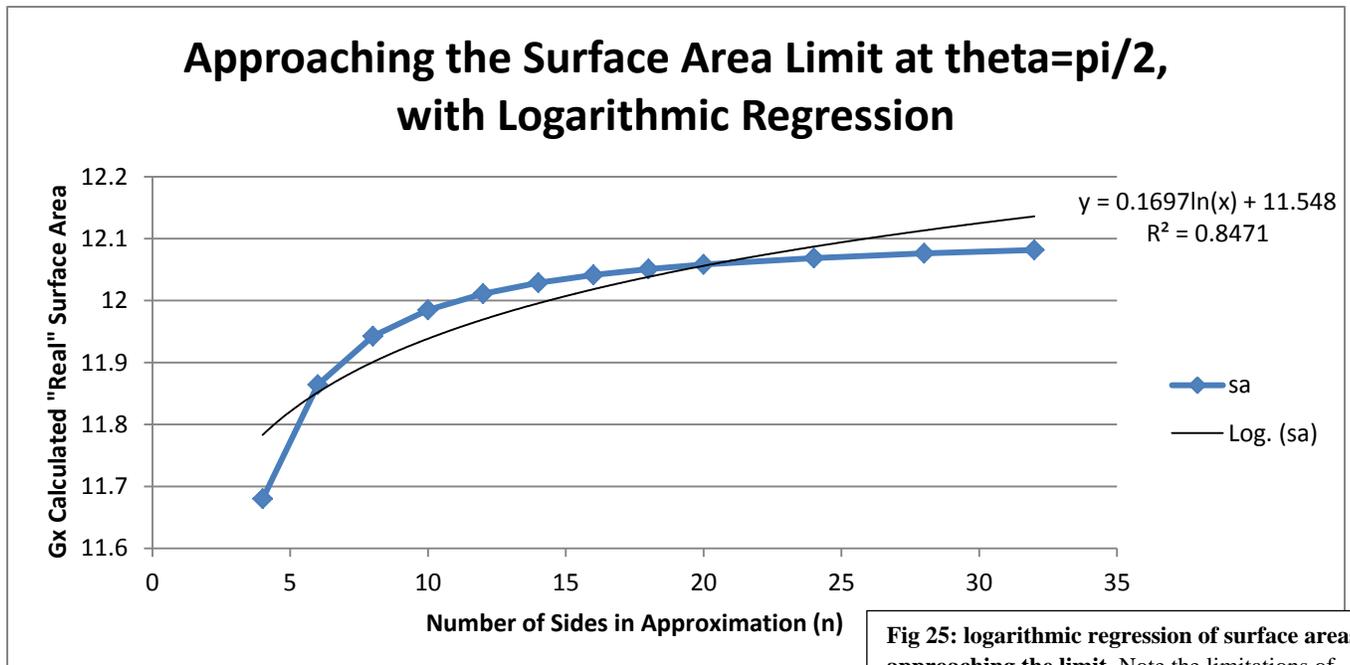


Fig 25: logarithmic regression of surface areas approaching the limit. Note the limitations of the model, resultant of the lack of a horizontal asymptote being approached by a log function.

problem with using a logarithmic regression here is that it doesn't approach a limit; this leads to it being unsuccessful in approximating the curve at the upper extreme of n , making it of little use. The regression was calculated in Microsoft Excel, and then confirmed with a TI-89 calculator.

All of these representations of the limit show that the 32 side approximation is essentially close enough to the actual area to be credible to about the hundredths place. This entire process need not be repeated for every twist angle and height to radius ratio; because of the proven accuracy of the 32 side approximation, it can be used to find the surface area for whatever parameters are set forth.

Discussion/Conclusion:

Discovering and modeling the squeeze in the Twisted Savonius VAWT as it twists has led to the important knowledge of exactly how to build the turbine in terms of geometric parameters. The newfound understanding of the geometric constraints of this complex shape can be combined with the known and yet-to-be-known constraints of the materials used to build it to create an improved turbine that is both cheaper to produce and more efficient in operation. Restraints in twisting that extended beyond materials previously are now explained by this theory of *geometric squeezing* of the shape. The squeeze's modeling has also simplified the construction method, by limiting the sections of true Savonius shape to four, instead of approaching the limit of infinity. Previously, many turbines were carefully molded into the "true" twisted Savonius shape where *every* horizontal cross section had the same radius. This is no longer necessary as the squeeze has been shown to be minimal, with maximum twist and construction efficiency, when the turbine is twisted $\pi/4$ radians, or 45° between each of the five true cross sections, which can act as supports.

One piece that could be overlooked is the visuals that were created using the pseudo-three-dimensional model animations. These allow for visualization of the turbines in operation and can be layered in *Geometry Expressions* to show what the turbines would look like on a building from virtually any angle (street or air) in any colors, even semi-transparent; but most importantly *in operation*. Figure 26 shows a still version of this, with the building being the Oregon Episcopal School (OES) dining hall. Knowing the surface area of the blade also allows cost estimates to be made much more accurately. The most important surface area figure is that when there is an overall a:d (height to diameter) ratio of 2:1 (individually .5:1) and the twist is $\pi/4$ radians for each section of each side, which is $\sim .82$ units squared. Therefore, the total surface area for the fully twisted model at the simplest logical a:d ratio is ~ 3.28 units squared, a useful figure to know for efficiency and material costs purposes.

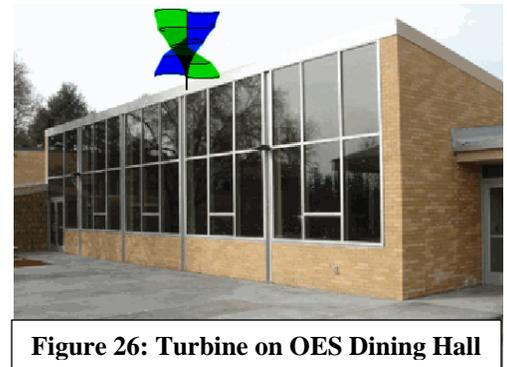


Figure 26: Turbine on OES Dining Hall

There are several limiting factors, however. The first issue is that the visual models are technically two-dimensional and depict a three dimensional shape. The use of ellipses and semi-transparent blades makes this less noticeable, but when viewing the animation of the turbine “in operation” the turbine can appear to be spinning either clockwise or counter-clockwise, depending on how you look at it, even though it is mathematically spinning clockwise. The triangle approximations are also *approximations* of the actual shape and many more triangles may be needed to better this approximation. One positive is that the angle of 45° was found by a previous study (Mehdi, 2006), to be the most efficient, which is now also demonstrated geometrically. Saha (2005) stated that more twist is better, which is also true in principle, and the stacking method allows higher twist angles to be used.

All of the data collected does allow cost estimates to be made, though exact figures vary widely based on materials. A 2 meter tall blade (for a 3 meter turbine) could be built from the triangle approximations for under \$50 with the axle/mount, but without gears, generator, and electronics. I predict that a scaled up version of the 32 side approximation could be cut out of semi-flexible fabric and stretched over the frame, then coated with something to fill any small holes, all to create a very simple, cheap and effective blade. This compares to current costs that can range up to \$15,000-\$25,000 for the *Aerotecture* “Aeroturbine”, figure 27, which is a similar design (Aerotecture, 2010 [and fig. 27]). There are still many limitations to the triangle approximation models. The biggest is that because they are approximating a complex shape, more triangles could always be used to make the models. Additionally, a refined method for utilizing the triangle approximation models, without making all of the folds along each triangle, needs to be developed before the model is fully ready for use. Overall, however, the cheaper cost and simpler construction open the door for widespread use of the Twisted Savonius design methods in rooftop settings. With this simpler design and construction method, they could feasibly and inexpensively be placed on all suitable buildings (see Appendix E) to generate large amounts of power.



Figure 27: Aerotecture’s “Aeroturbine”

List of Citations

- (2005-2006). Discover Solar Energy ADVANTAGES DISADVANTAGES|PROS and CONS of Solar Energy. Retrieved July 23, 2009, from facts-about-solar-energy.com Web site: <http://www.facts-about-solar-energy.com/solar-energy-advantages-disadvantages.html>
- (2008, 7 23). Savonius wind turbines--Wind. Retrieved August 4, 2009, from REUK.co.uk Web site: <http://www.reuk.co.uk/Savonius-Wind-Turbines.htm>
- (2009). Wind turbines noise and vibration|Wind turbines. Retrieved August 13, 2009, from Bettergeneration.com Web site: <http://www.bettergeneration.com/wind-turbines/wind-turbines-noise-and-vibration.html>

(2009, 7 7). Darrieus wind turbine--Wikipedia, the free encyclopedia. Retrieved July 31, 2009, from Wikipedia.org Web site:

http://en.wikipedia.org/wiki/Darrieus_wind_turbine

(2009, 7 7). Savonius wind turbine--from Wikipedia, the free encyclopedia. Retrieved August 4, 2009, from Wikipedia.org Web

site: http://en.wikipedia.org/wiki/Savonius_wind_turbine

(2009, 7 9). Vertical axis wind turbine. Retrieved July 26, 2009, from Wikipedia Web site:

http://en.wikipedia.org/wiki/Vertical_axis_wind_turbine

(2010). Aerotecture International - 510V Aeroturbine. Aerotecture International - Home. Retrieved February 21, 2011, from

http://www.aerotecture.com/products_712V.html

(No date (n.d.)). *Hydroelectric Power – Water Power – micro hydro systems*. Retrieved January 31, 2009, from

<<http://www.green-trust.org/hydro.htm>>.

Blocken, B. (n.d.). *Numerical simulation of wind flow around buildings*. Retrieved from http://www.ates.co.kr/e&c/data/3_3.pdf

Energy Information Administration (2009, July). *Average Retail Price of Electricity to Ultimate Customers: Total by End-Use*

Sector. Retrieved July 22, 2009, from http://www.eia.doe.gov/cneaf/electricity/epm/table5_3.html

Halsey, N. (2010, 3). *Rooftop Wind Turbines: Where Should They be Placed and What Design is Best?* Unpublished. Oregon

Episcopal School, Portland, OR.

Hussain, M., Mehdi, S. N., & Reddy, P. R. (2008, January 1). CFD analysis of low speed vertical axis wind turbine with twisted

blades. *The free library*. Retrieved August 18, 2010, from

<http://www.thefreelibrary.com/CFD+analysis+of+low+speed+vertical+axis+wind+turbine+with+twisted...-a0216041119>

In the Public Interest: How and Why to Permit for Small Wind Systems. (2008, 9). *A Guide for State and Local Governments*.

Retrieved October 18, 2010, from <http://www.awea.org/smallwind/pdf/InThePublicInterest.pdf>

Oke, C. (2010). *Saltire Software*. Retrieved from <http://www.saltire.com/>.

Oke, C. (2010, August 17). *Geometry Expressions*. Retrieved from

<http://www.geometryexpressions.com/>

Saha, U. K. & Rajkumar, M. J.. (2005, August 6). On the performance analysis of Savonius rotor with twisted blades.

sciencedirect.com. Retrieved August 18, 2010, from

www.victordaniilochkin.org/research/turbine/papers/On%20the%20performance%20analysis%20of%20Savonius%20rotor%20with%20twisted%20blades.pdf

Rooftopsolar2. (2010). [Web]. Retrieved from <http://www.brisk solar.com/solar-roof->

[top-panels.php](http://www.brisk solar.com/solar-roof-top-panels.php)

Van Den Berg, G. P. (2004, 9 4). Do wind turbines produce significant low frequency sound levels?. Retrieved August 13, 2009, Web site: http://www.viewsofscotland.org/library/docs/LF_turbine_sound_Van_Den_Berg_Sep04.pdf

Acknowledgements

I would like to acknowledge the following people for their assistance in this project:

- **Phillip Todd**, Mentor, founder of Saltire Software, creator of *Geometry Expressions*
- **Jim Wiechmann**, Co-Mentor, HS Math Teacher and part time employee of Saltire Software
- **Chris Oke**, *Geometry Expressions* programmer
- The rest of the staff of **Saltire Software**
- **Jack Turzillo, Katie Purdy**, co-interns, random mathematical assistance

Appendices

Appendix A: List of Geometry Expressions Features.

Geometry Expressions was originally published in 2006 and is used primarily as an educational tool at high school and college levels of mathematics classes (Algebra and Geometry through advanced Calculus). The program has the following list of features (as of latest released version 2.2):

- **Drawing Shapes:** You can draw and constrain the following objects by using icons in the Draw tool panel:
 - Points
 - Line Segments
 - Infinite Lines
 - Vectors
 - Polygons
 - Circles
 - Conics:
 - Ellipse
 - Parabola
 - Hyperbola
 - Arcs (on circles and conics)
 - "N-gons" (Regular polygons of N sides)
 - Curve Approximations
 - Functions
 - The Draw tool panel also inserts text, pictures and expressions.
- **Constraints:** You can make the following constrains by using icons in the Constrain (Input) tool panel:
 - Length/Distance
 - Radius
 - Perpendicular
 - Angle

- Direction
 - Slope
 - Coordinates
 - Coefficients
 - Tangent
 - Incident
 - Congruent
 - Parallel
 - Implicit Equation
 - Point Proportional along curve
 - You can only make these constraints on the appropriate objects. For example, you can't constrain a line segment's radius. The software automatically displays only the logical options for the item(s) selected to simplify the process for the user.
- **Constructions:** You can construct a variety of objects in *Geometry Expressions*. Constructions differ from constraints because they create more objects while constraints change the positioning of existing objects. The following constructions are available from the Construct tool panel:
 - Midpoint
 - Intersection
 - Perpendicular
 - Perpendicular Bisector
 - Angle Bisector
 - Parallel
 - Perpendicular
 - Tangent
 - Polygon (this is especially useful because it can include arcs)
 - Transformations:
 - Reflection
 - Translation
 - Rotation
 - Dilation (scaling)
 - Locus
 - Trace (of locus, curve, etc. along a proportional point)
 - As with constraints, you can only make these constructions on the appropriate objects. For example, you can't construct a tangent to a line segment. Again, the software automatically displays only the logical options for the item(s) selected to simplify the process for the user.
- **Calculations:** You can make many of the same calculations as the constraints, with the addition of things like area and perimeter. Calculations can be made in both symbolic and real notation so when you use variables your calculation can be in terms of those variables or a decimal of the numbers represented by the variables. When not using variables, symbolic calculations give an exact output and real calculations give an approximate output. Here is a complete list of the available calculations:
 - Length/Distance

- Radius
- Angle
- Direction
- Slope
- Coordinates
- Coefficients
- Area
- Perimeter
- Parametric Equation
- Implicit Equation
- Like with the other features, calculations these can be made only on the logical objects, so the area of a line cannot be calculated.
- **Variables:**
 - When constraints are made symbolically, *Geometry Expressions* can drag or even animate variables that are incorporated in the constraints. Functions can also be input symbolically in constraints, and then changed from the variables tool panel.
- **Other:**
 - There is also an annotation feature and a Symbols tool panel which inserts Greek letters, exponents, fractions and other common functions (which can also be written out manually).
- **Formatting:**
 - Control of color, style, thickness and transparency of all objects drawn/constructed
 - Option to show with or without axis, with or without a grid
 - Ability to show/hide all drawn objects and toggle between shown and hidden

Appendix B: A Geometric Proof discovered through investigation of the top view model (see also project website: <http://www.saltireserver.com/gx/vawt/content/4.%20A%20Geometric%20Proof.html> for a more interactive version):

Proof that angle HEG in figure 1 has the measure of theta

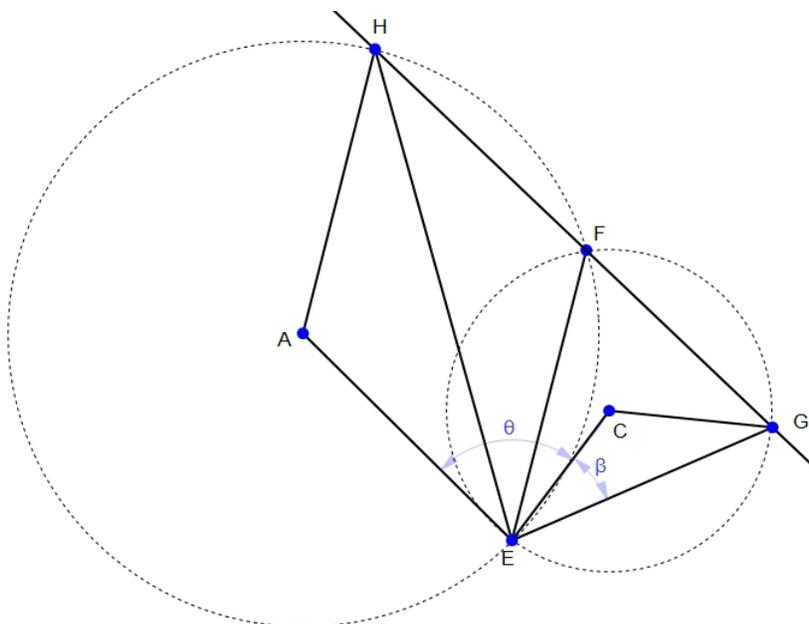


Figure 1—the base figure:

Here we have two circles whose center points are A and C and whose intersection points are E and F. Points G and H are on circles C and A respectively. We have constrained line HG to go through point F and angles AEC and CEG are constrained as theta and beta. Our objective

is to prove that angle HEG is always theta also when the above conditions are met.

Geometry Expressions Proof:

Draw the figure with all of the above constraints. Calculate angle HEG. If the output is theta, then (assuming *Geometry Expressions* is correct) we have proven that the angle is theta.

{See video 1—proving with Geometry Expressions online}

Formal/traditional Proof:

To prove that angle HEG is theta, we need to first prove that angle AEH is beta. To do this, we can first show that triangles HAE and GCE are isosceles because two of their sides are radii and the third is a chord. Therefore, angle CGE is also beta and angle ECG is $2\pi - \beta$ because of the triangle angle sum theorem. Then the chord angle theorem tells us the value of angle EFG. {see video 2—the chord angle theorem} In this case, we can now show that angle EFG is $\pi/2 - \beta$. We can also see that angle EFH is $\pi/2 + \beta$. See figures 2 and 3.

Figure 2—chord angle theorem:

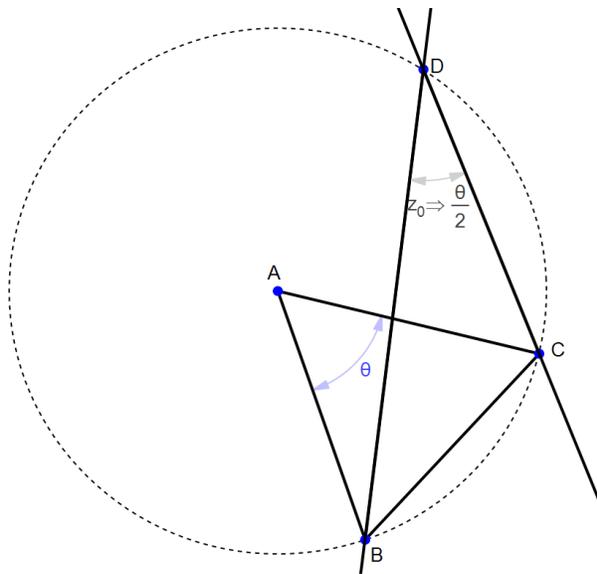
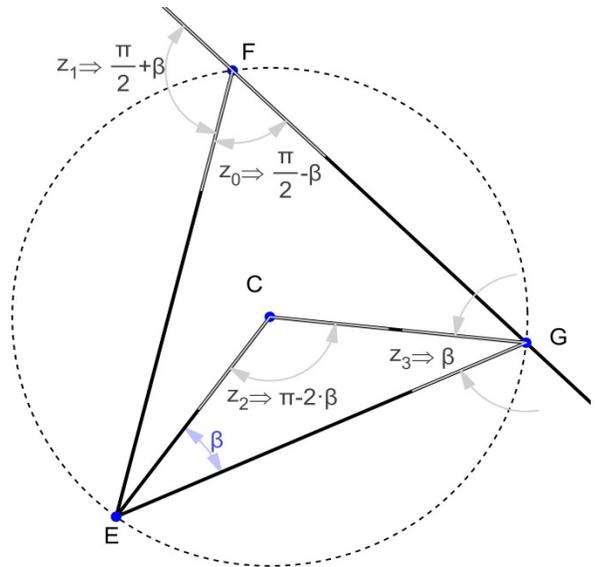


Figure 3—circle C angles:



Now, we have an angle in circle A, so we are very close to proving that angle AEH is beta. We need to use another theorem, the cyclic quadrilateral theorem, figure 4 {see video 3—the cyclic quadrilateral theorem}. Figure 5 shows its relation to this context. Then we combine this with the chord angle theorem from before to find angle HAE. Because the triangle is isosceles, we have proven that angle AEH is beta (figure 6). Now we can take either of the two following paths.

Path 1:

We know through the addition of angles AEC and CEG that angle AEG is equal to beta+theta and that angle AEH is beta, so angle HEG must be theta to satisfy that angle AEG is beta+theta (figure 7).

Path 2:

Angles AHE and CGE are also beta (because of the isosceles triangles) so triangles HAE and GCE are similar through AA. We can see that their relation is through a rotation about point E as well as a size reduction. Since the corresponding sides AE and CE have an angle given as theta between them, the other corresponding sides HE and GE also have an angle of theta between them; so angle HEG is equal to theta (figure 7).

Figure 4—the cyclic quadrilateral theorem:

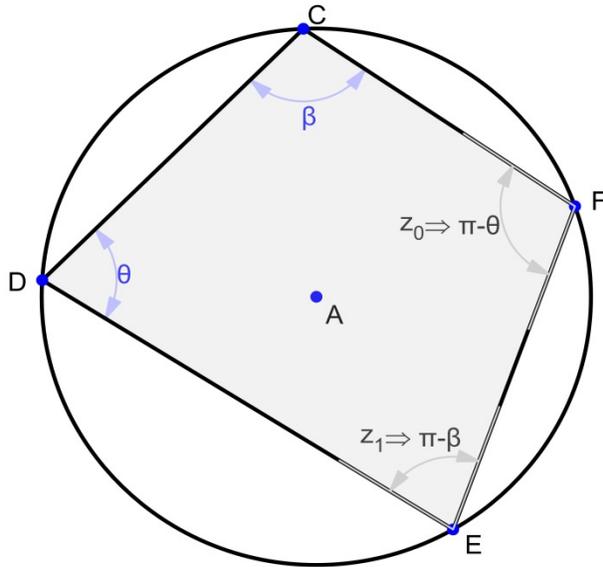


Figure 5—our cyclic quadrilateral:

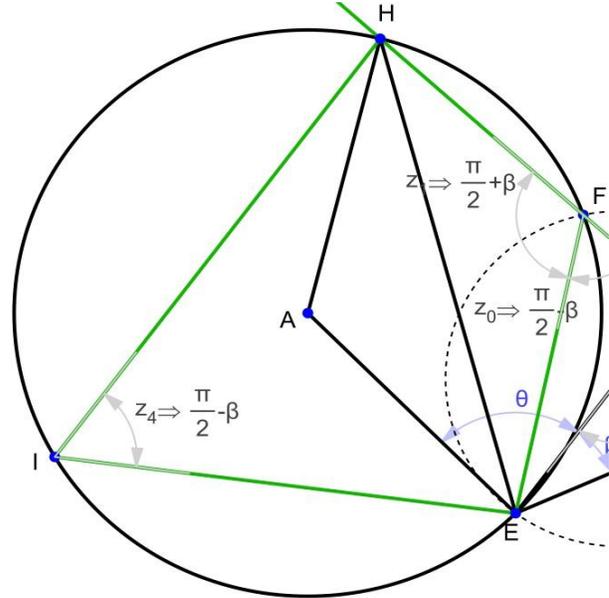


Figure 6—circle A angles:

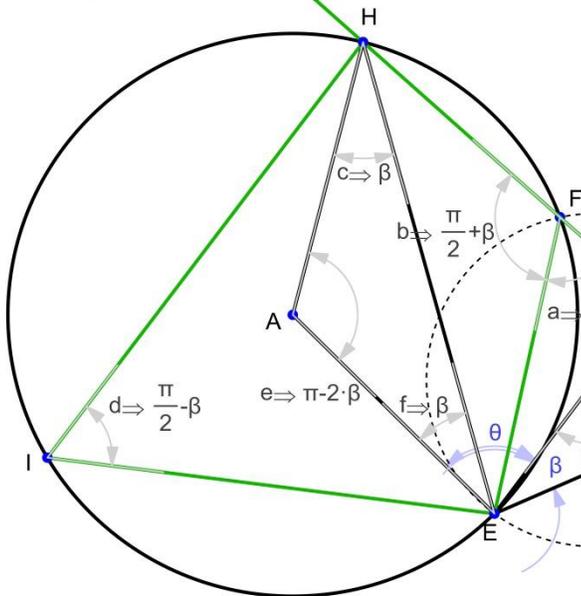
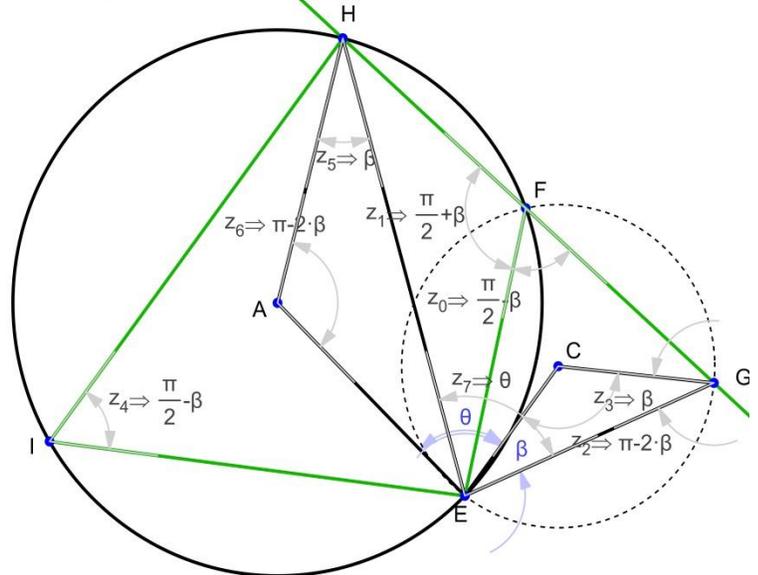


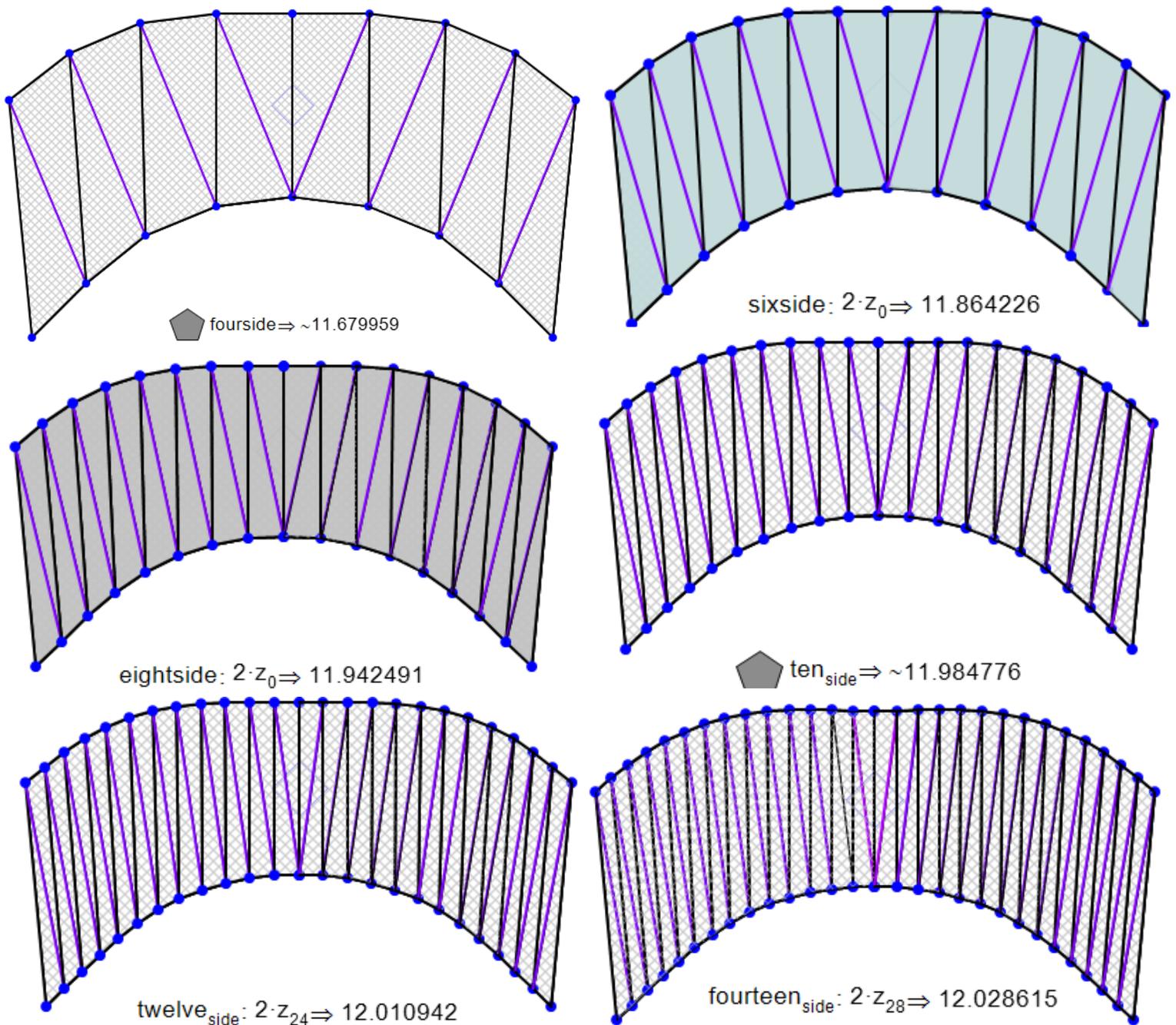
Figure 7—final figure:

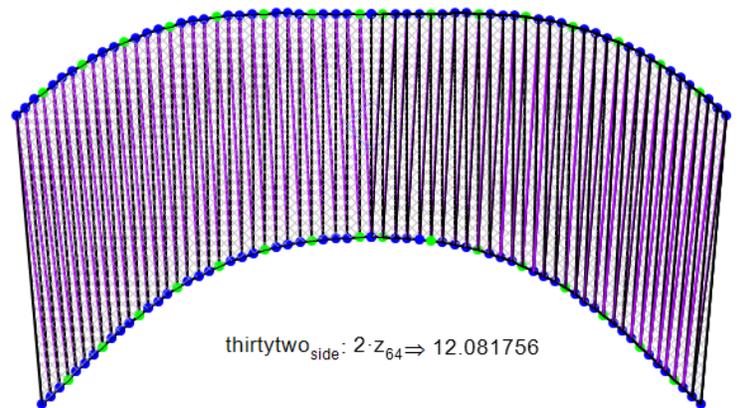
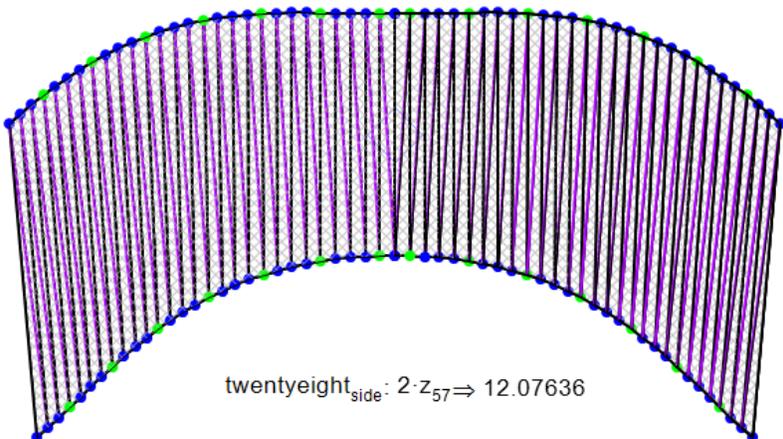
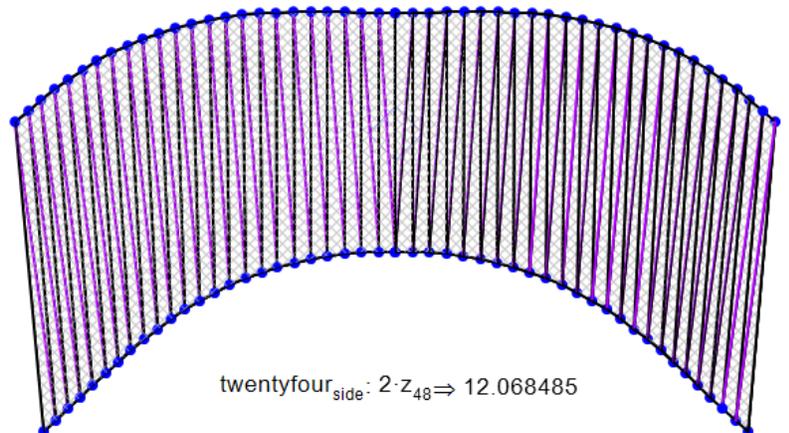
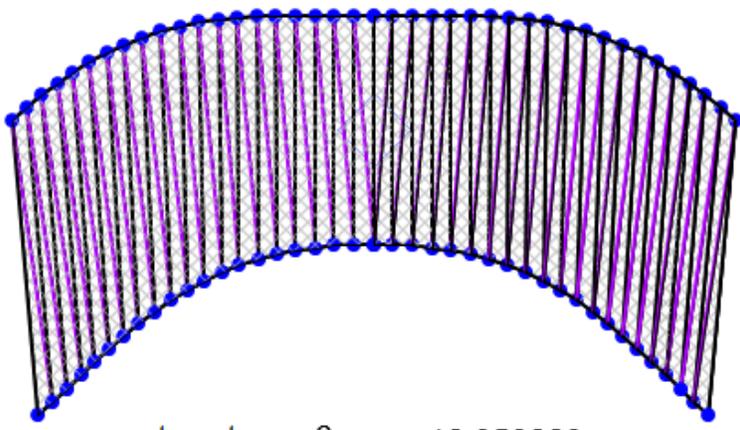
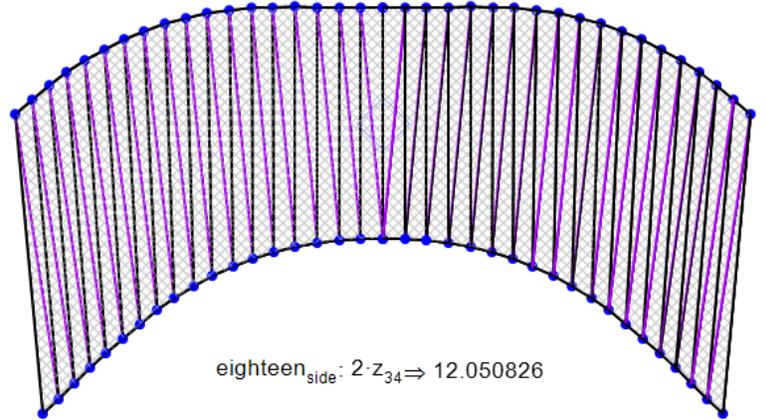
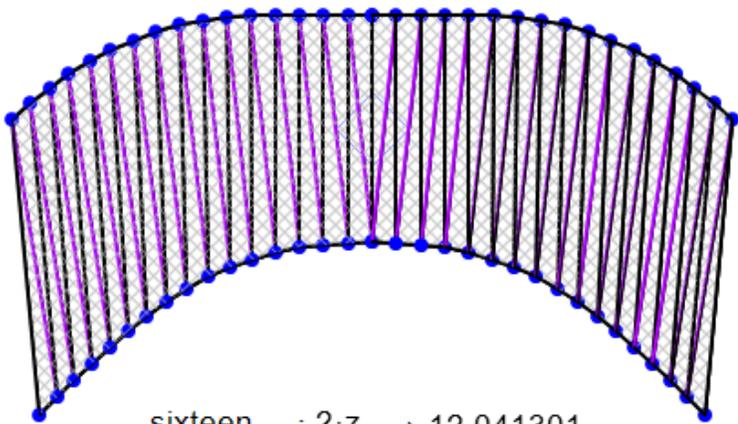


Appendix C: Bibliographic notes:

This project can also be found online at www.saltireserver.com/gx/vawt/index.html. This page is hosted by Saltire Software, the maker of *Geometry Expressions* and is among other student projects centered around *Geometry Expressions*. All of these, including this one are found at [http://www.geometryexpressions.com/explore.php?p=03-Student Projects](http://www.geometryexpressions.com/explore.php?p=03-Student%20Projects). Figures from this project can also be found in various locations on Wikipedia, including the Geometry Expressions Wikipedia page, and most notably on the general Wind Turbine page (in several languages) which hosts an animated version of figure 4 from this paper.

Appendix D: All of the unrolled triangle approximation models with twists of $\pi/4$ radians and an overall a:d ratio of 1.83:1, as used by prior researchers, and an individual a:r radius of .46:.25:





Appendix E: An OES campus satellite image map (Google Earth Imagery) with potential turbine locations indicated in green and blue based on building height, shape and orientation considerations. It demonstrates the potential for widespread usage of wind turbines in rooftop applications.

Future considerations for rooftop wind turbines at OES:

