Using Spherical Coordinates to Calculate Center GPS Coordinates of Center Pivot Sprinklers

Dane Thompson
Electrical Engineering
Kansas State University

4 May 2022

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RTK………………………………………………………………… Real-Time Kinematic

GPS………………………………………………………………… Global Positioning System

1. EXECUTIVE SUMMARY

Many companies produce GPS monitors for center pivot irrigation sprinklers. These monitors allow farmers to remotely view the position of the sprinkler in the field. To represent the position of the sprinkler, a line is drawn from the center point of the sprinkler to the current location of the sprinkler's end tower. Most remote sprinkler monitoring solutions require the farmer to manually select the center point of the sprinkler.

Instead of manually selecting the center point of the sprinkler, I propose calculating the center point of the sprinkler from a set of the sprinkler's end tower points. Trimble, Inc. has already patented a method for calculating the center point of a sprinkler. However, I have designed an alternative method that also calculates the center point of a sprinkler. The fundamental difference between Trimble's method and mine is that my method carries out all calculations in 3-dimensional coordinates, while Trimble's method includes a conversion from 3-dimensional coordinates to 2-dimensional coordinates. For reference, GPS coordinates are 3-dimensional. I simulated my design against Trimble's for three metrics: efficiency, accuracy, and ability to scale with field size.

I found that my method was 7 times more efficient than Trimble's. When each of our methods were used on a given number of data points, Trimble's method took 7 times longer to calculate the center point than my method.

Accuracy was tested for two scenarios: perfect GPS points and deviated GPS points. When the GPS points were perfectly around a circle, Trimble's method had better accuracy than my method. However, my method was still accurate within a couple meters. When the GPS points included a deviation of 10 meters, my model was more accurate than Trimble's. These results are in favor of my method, as standalone GPS is only accurate to 10 meters. Almost perfect accuracy can be obtained by real-time kinematic (RTK) GPS but is more expensive. Overall, both methods had sufficient accuracy for this application using standalone GPS.

One of Trimble's method's biggest issues is its inability to scale with the size of the field. Most fields are small enough that Trimble's method is sufficient, but I wanted to push both methods to the extreme. I found that my method was able to scale much better for larger fields than Trimble's method. For example, when calculating the center of an imaginary field with a 100-mile radius, Trimble's method calculated a center point 1245.6 feet from the actual center while my method calculated a center point only 64 feet away.

2. INTRODUCTION

This report describes a method of calculating the center GPS coordinates of a center pivot sprinkler using the spherical coordinate system. Remote sprinkler position monitoring is a useful tool for farmers to be able to check the location of their sprinkler in the field from their phone or computer. To represent the position of the sprinkler, two locations must be known: the center coordinates of the sprinkler and the current end tower coordinates. Once these locations are known, a straight line is drawn between them to represent the sprinkler. Traditionally, a remote monitoring unit sits on the end tower of the sprinkler to broadcast the current end tower coordinates. The center coordinates are usually hardcoded into the monitor, such as with Lindsay's FieldNET monitors [1], or an additional monitoring unit may be attached to the center tower of the sprinkler as described in Reinke Manufacturing's patent [2]. Instead of using these two methods for the center coordinates, I propose calculating the center coordinates of the sprinkler from the coordinates collected from the end tower unit. This would result in a true plug-and-play experience for farmers. A method for calculating the center coordinates of a sprinkler already exists, and I will be using it as a baseline for comparison against my proposed method.

In this report, I will describe the existing method of center coordinate calculation and its drawbacks and then describe my proposed method. Once both methods have been defined in my report, I will compare their computational complexity, accuracy, and use in various sized fields. The comparisons will be made quantitatively.

The information I gathered for this report is from published patents, published technical literature, and reputable online sources. Many of the comparisons I made include empirical data that I produced from my own simulations. Each simulation is described in the report, and I attempted to make them fair for each method.

The rest of my report is split into three main sections. First, in Section 3, I will discuss the existing method for calculating the center coordinates of a center pivot sprinkler. Then, in Section 4, I will describe and derive my method for calculating the center coordinates. Finally, in Section 5, I will directly compare the performance of each method through simulation.

3. CENTER CALCULATION USING CARTESIAN COORDINATES

One method of calculating the center coordinates of a center pivot sprinkler was designed by Trimble, Inc [3]. In this section I will describe their method, its drawbacks, and their algorithm's assumptions that cause these drawbacks.

Their algorithm begins by taking a set of 3-dimensional GPS points and converting them to 2-dimensional Cartesian coordinates. While Trimble does not specify how this conversion takes place, I am assuming that they are using the Haversine formula. The Haversine formula finds the distance along the surface of a sphere between two points on the surface of the sphere [4]. After arbitrarily choosing one of the GPS points in the set as a reference origin point, Trimble's method presumably uses the Haversine formula to calculate the horizontal (x) and vertical (y) distances for each point relative to the origin. After calculating a set of (x, y) points around the circular arc of the center pivot sprinkler, they run a linear regression on the points using the following circle equation:

$$(x-h)^2 + (y-k)^2 = r^2 \tag{1}$$

I will discuss how linear regression works later in this report, but for now it is only important to know that the regression will find values for h, k, and r. The value r represents the radius of the 2-dimensional circle while values h and k represent the center point of the 2-dimensional circle, $P(h, k)$ in Cartesian coordinates. After finding the 2-dimensional center, Trimble's method then converts the 2-dimensional center point back into 3-dimensional GPS points presumably using the Haversine formula in reverse.

3.1. Drawbacks

This section discusses two negative effects of using Trimble's method: required knowledge of the field elevation and the inability to scale for field size.

3.1.1. Required to Consider Elevation

One of the issues with Trimble's Cartesian method is that knowledge of the field's elevation is required. While a quick online search yields Earth's radius as 3,963 miles, this radius is calculated at sea level [5]. Therefore, Trimble must add the elevation of the field above sea level to the value of Earth's radius in their calculations. Most GPS units provide elevation data in addition to coordinates, so this is possible, but it is another piece of data that Trimble's method must consider.

3.1.2. Scaling Issues with Larger Fields

The biggest problem with Trimble's Cartesian method is that its accuracy breaks down as the size of the field increases. The standard size of a field is a quarter-section or 160 acres. Fields larger than this will usually have multiple smaller sprinklers instead of one large sprinkler. However, center pivot sprinkler manufacturers could design larger pivots in the future. In Table 1, below, the accuracy of Trimble's method is compared against the size of the field where the center is being calculated. Each field has 1000 ideal GPS points chosen along a circular arc of the given radius.

Table 1. Effect of field size for Trimble's method of center coordinate calculation

Radius of Field	Distance to Center Point
0.25 miles	0.01 feet
1 mile	0.15 feet
5 miles	3.4 feet
10 miles	13.4 feet
20 miles	52.9 feet
100 miles	1245.6 feet
200 miles	4633.0 feet

As shown above in Table 1, the accuracy of Trimble's method decreases as the size of the field increases. For most smaller-sized fields, Trimble's method is sufficient. However, as the field size increases, Trimble's method is no longer able to accurately find the center point. Trimble's method will work for calculating the center point of a center pivot sprinkler in most cases but does not have the ability to scale with field size. Thus, Trimble's method will most likely not have any other applications besides center pivot sprinklers.

3.2. Causes

Both issues stem from the conversion of 3-dimensional GPS coordinates to 2-dimensional Cartesian coordinates. This section will describe the mathematical reasoning for requiring field elevation and discuss Trimble's assumption of a flat field.

3.2.1. Coordinate Conversion Using the Haversine Formula

Trimble's method is required to consider the elevation of the field because of the use of the Haversine formula when converting from 3-dimensional GPS coordinates to 2-dimensional Cartesian coordinates. The Haversine formula is given by the following equation [4]:

$$haversin\left(\frac{distance}{radius}\right) = haversin(Lat_2 - Lat_1) + \cos(Lat_1)\cos(Lat_2)\, haversin(Lon_2 - Lon_1) \qquad (2)$$

This equation relates the distance between two GPS coordinates, (Lat_1, Lon_1) and (Lat_2, Lon_2), to the radius of the Earth. However, the radius is referring to the radius of the Earth at the location of the coordinates. Since elevation can vary from location to location, Trimble's method must account for this elevation in practice.

3.2.2. Ignoring Curvature of Earth

Trimble's field-size scaling issue stems from their method's assumption that the field is flat. For smaller fields, this assumption can be made without inducing much error. However, as the size of the field increases, the curvature of the field increases. This curvature is described in Figure 1.

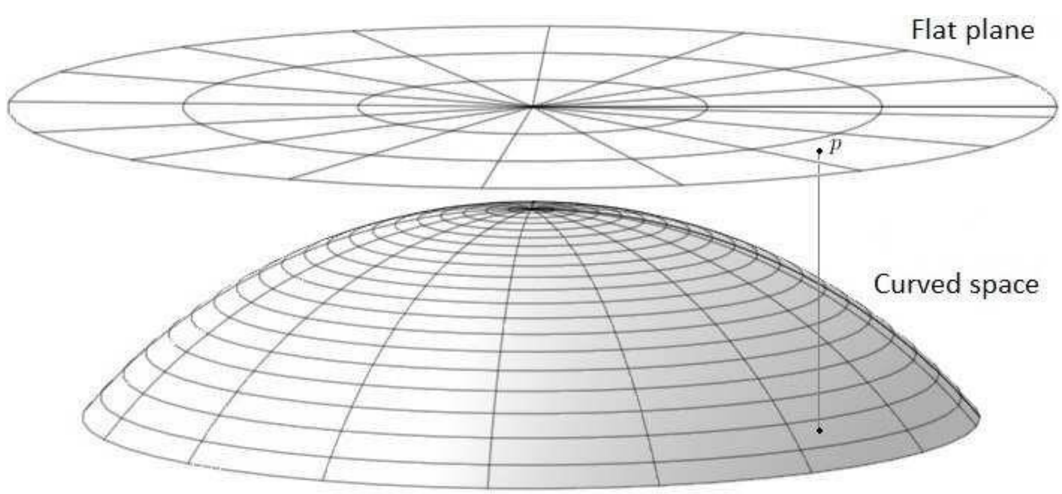


Figure 1. Flat plane cast onto a curved space. (Reproduced from [6])

Trimble's method takes points along a curved space, like the one at the bottom of Figure 1, and assumes that the curvature is small enough that the space looks like the flat plane at the top of Figure 1. This assumption can be made for smaller fields, such as a standard 160-acre field. The assumption eventually breaks down because the reference origin point of the 2-dimensional Cartesian coordinates is arbitrarily chosen and not centered. If the origin was centered relative to the set of Cartesian coordinates, I hypothesize that Trimble's method would work. However, the whole purpose of Trimble's method is to calculate the center, so it is impossible to choose an origin that is centered relative to the set of coordinates.

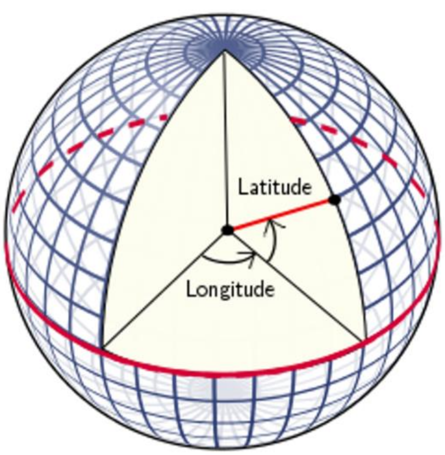
4. CENTER CALCULATION USING SPHERICAL COORDINATES

To avoid the issues in Trimble's model, I designed an alternative to calculate the center coordinates of the sprinkler that does not require conversion to the Cartesian coordinate system. In this section, I describe my alternative by giving a brief description of GPS coordinates, describing a geometric analogy, mathematically deriving my algorithm, and simulating it in MATLAB.

4.1. GPS Coordinates Overview

GPS coordinates come in latitude and longitude pairs as described on the left in Figure 2.

GPS Coordinates versus Spherical Coordinates



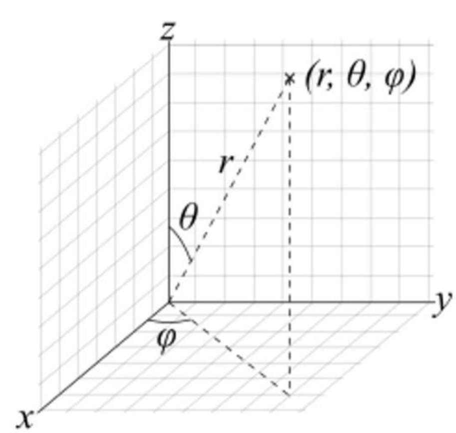


Figure 2. Visual Representation of GPS Coordinates (Left, reproduced from [7]) **and Visual Representation of Spherical Coordinates** (Right, reproduced from [8])

The coordinate system used by GPS on the left is almost identical to the spherical coordinate system which is displayed on the right in Figure 2. The only difference is the spherical system uses an inversion and 90° offset of the GPS latitude. Thus, GPS coordinates and spherical coordinates can be related with Equations (3) and (4) [7]:

$$\theta = 90° - Latitude_{GPS} \tag{3}$$

$$\varphi = Longitude_{GPS} \tag{4}$$

The equations above show that GPS coordinates and spherical coordinates are practically identical. Therefore, I will use the terms "spherical coordinates" and "GPS coordinates" interchangeably in the derivation of my solution.

4.2. Cone Inscribed Sphere

The first step I took to design my solution was to consider the geometry of the problem. It seemed obvious to represent the Earth as a sphere, but I needed to get creative for how I would represent the circular arc of the center pivot sprinkler. Figure 3, below, depicts a cone inscribed in a sphere.

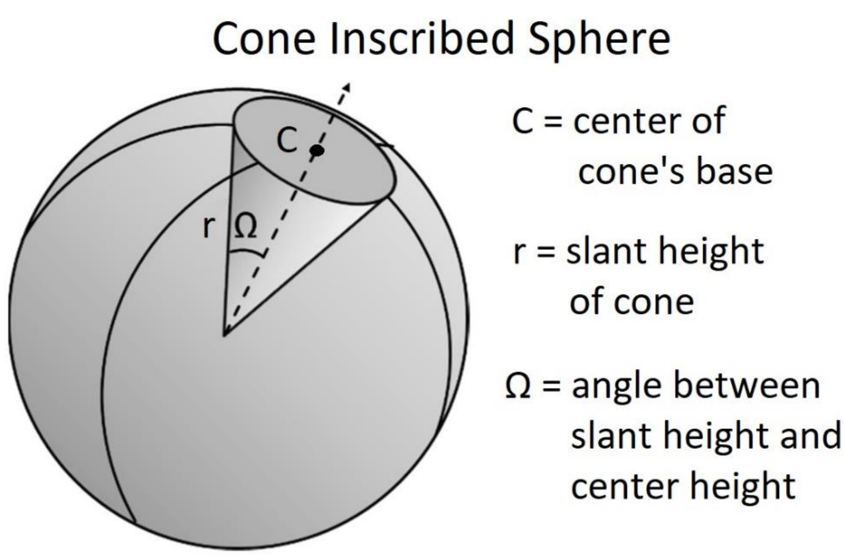


Figure 3. A Cone Inscribed in a Sphere (Adapted from [9])

My solution using the spherical coordinate system relies on the geometry displayed above in Figure 3. First, I considered two shapes: a sphere representing the earth and a cone whose base represents the area of a field covered by a center pivot sprinkler. The apex, or point, of the cone is located at the center of the Earth and the slant height r represents the radius of the earth. The angle

Ω is the angle between the cone's slant height and center height. Angle Ω is constant around the cone, thus the angle between any point along the cone's base perimeter and center point C is angle Ω. Knowing this fact, I can construct a regression model that takes points along the cone's base perimeter as input and attempts to find a center point C where the angle between point C and each base perimeter point is constant.

4.3. Vector Dot Product

The next step in designing my solution was constructing a regression model. I knew that I needed a model that related angle Ω to the center coordinates and coordinates of the cone's base perimeter. The equation I decided on was the vector dot product.

The vector dot product is defined by (5) for two Cartesian vectors $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ extending to points $P(A_x, A_y, A_z)$ and $P(B_x, B_y, B_z)$, respectively:

$$\text{where } \overrightarrow{V_1} = A_x\hat{x} + A_y\hat{y} + A_z\hat{z} \text{ and } \overrightarrow{V_2} = B_x\hat{x} + B_y\hat{y} + B_z\hat{z}$$

$$\overrightarrow{V_1} \cdot \overrightarrow{V_2} = A_xB_x + A_yB_y + A_zB_z = \left|\overrightarrow{V_1}\right|\left|\overrightarrow{V_2}\right| cos\Omega \tag{5}$$

It should be noted that Ω is the angle between vectors $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ and the result of their dot product is a scalar number. Equation (5) includes angle Ω but does not yet include the center coordinates or coordinates of the cone's base. To include these parameters, I can make the following substitutions:

$$\text{where point } P\left(A_x, A_y, A_z\right) = P(r_1, \theta_1, \varphi_1) \text{ and point } P\left(B_x, B_y, B_z\right) = P(r_2, \theta_2, \varphi_2)$$

$$A_x = r_1 sin\theta_1 cos\varphi_1 \text{ and } B_x = r_2 sin\theta_2 cos\varphi_2 \tag{6}$$

$$A_y = r_1 sin\theta_1 sin\varphi_1 \text{ and } B_y = r_2 sin\theta_2 sin\varphi_2 \tag{7}$$

$$A_z = r_1 cos\theta_1 \text{ and } B_z = r_2 cos\theta_2 \tag{8}$$

By substituting (6), (7), and (8) back into the previous dot product equation, I get the following equation:

$$\overrightarrow{V_1} \cdot \overrightarrow{V_2} = r_1 sin\theta_1 cos\varphi_1 r_2 sin\theta_2 cos\varphi_2 + r_1 sin\theta_1 sin\varphi_1 r_2 sin\theta_2 sin\varphi_2 + r_1 cos\theta_1 r_2 cos\theta_2 \tag{9}$$

Next, I can factor out radii r_1 and r_2:

$$\overrightarrow{V_1} \cdot \overrightarrow{V_2} = r_1 r_2 (sin\theta_1 cos\varphi_1 sin\theta_2 cos\varphi_2 + sin\theta_1 sin\varphi_1 sin\theta_2 sin\varphi_2 + cos\theta_1 cos\theta_2) \quad (10)$$

The magnitudes $\left|\overrightarrow{V_1}\right|$ and $\left|\overrightarrow{V_2}\right|$ are equal to r_1 and r_2, respectively. Therefore, the dot product in terms of the magnitudes of $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ is the following:

$$\overrightarrow{V_1} \cdot \overrightarrow{V_2} = \left|\overrightarrow{V_1}\right|\left|\overrightarrow{V_2}\right| cos\Omega = r_1 r_2 cos\Omega \quad (11)$$

By substituting (11) into (10), I get the following equation:

$$r_1 r_2 cos\Omega = r_1 r_2 (sin\theta_1 cos\varphi_1 sin\theta_2 cos\varphi_2 + sin\theta_1 sin\varphi_1 sin\theta_2 sin\varphi_2 + cos\theta_1 cos\theta_2) \quad (12)$$

Notice that the product of radii r_1 and r_2, $r_1 r_2$, can be cancelled out of either side of (12). Therefore, (12) can be rewritten as:

$$cos\Omega = sin\theta_1 cos\varphi_1 sin\theta_2 cos\varphi_2 + sin\theta_1 sin\varphi_1 sin\theta_2 sin\varphi_2 + cos\theta_1 cos\theta_2 \quad (13)$$

Now, I have an equation that relates angle Ω and two spherical points $P(r_1, \theta_1, \varphi_1)$ and $P(r_2, \theta_2, \varphi_2)$. Next, I want to run a linear regression using Equation (13) as my model. Linear regression is the process of taking a set of data points and finding the optimal coefficients for a linear relationship. For example, if I have a set of (x, y) points and want to find the linear relationship between x and y, I can use the following regression model:

$$y = mx + b \quad (14)$$

The result of the regression is scalar values for variables m and b. Similarly, if I have a set of output y values with multiple inputs x_1 and x_2, the regression model (14) is adapted into the following:

$$y = m_1 x_1 + m_2 x_2 + b \quad (15)$$

Regression model (15) happens to have the same number of terms as Equation (13), so I will rewrite (13) to have the same input/output format as (15). First, I will divide both sides of (13) by $cos\theta_2$:

$$\frac{cos\Omega}{cos\theta_2} = \frac{sin\theta_1 cos\varphi_1 sin\theta_2 cos\varphi_2}{cos\theta_2} + \frac{sin\theta_1 sin\varphi_1 sin\theta_2 sin\varphi_2}{cos\theta_2} + \frac{cos\theta_1 cos\theta_2}{cos\theta_2} \quad (16)$$

Then, I will use the trigonometric identity (17) and move and reorder terms in (16) to get Equation (18) that will be used as our regression model:

$$tan\theta = \frac{sin\theta}{cos\theta} \quad (17)$$

$$cos\theta_1 = -tan\theta_2 cos\varphi_2 sin\theta_1 cos\varphi_1 - tan\theta_2 sin\varphi_2 sin\theta_1 sin\varphi_1 + \frac{cos\Omega}{cos\theta_2} \quad (18)$$

Equation (18) is referring to spherical points $P(r_1, \theta_1, \varphi_1)$ and $P(r_2, \theta_2, \varphi_2)$. Since r_1 and r_2 are not considered in this equation, the points can instead be stated as $P(\theta_1, \varphi_1)$ and $P(\theta_2, \varphi_2)$. To give more meaning to these two points, let's rename $P(\theta_1, \varphi_1)$ to $P(\theta_b, \varphi_b)$ and $P(\theta_2, \varphi_2)$ to $P(\theta_c, \varphi_c)$. Spherical point $P(\theta_b, \varphi_b)$ is referring to points along the base perimeter of the cone in Figure 3 and spherical point $P(\theta_c, \varphi_c)$ is referring to the center point C in Figure 3. The following equation is (18) with the new point definitions:

$$cos\theta_b = -tan\theta_c cos\varphi_c sin\theta_b cos\varphi_b - tan\theta_c sin\varphi_c sin\theta_b sin\varphi_b + \frac{cos\Omega}{cos\theta_c} \quad (19)$$

Note that (19) is the same form as (15) where:

$$y = cos\theta_b \quad (20)$$

$$m_1 = tan\theta_c cos\varphi_c \text{ and } x_1 = -sin\theta_b cos\varphi_b \quad (21)$$

$$m_2 = tan\theta_c sin\varphi_c \text{ and } x_2 = -sin\theta_b sin\varphi_b \quad (22)$$

$$b = \frac{cos\Omega}{cos\theta_c} \quad (23)$$

Recall that the goal is to calculate the latitude and longitude of the center of the sprinkler's arc. From the relationships defined in Equations (3) and (4), it should be clear that the latitude and longitude of the center is point $P(\theta_c, \varphi_c)$. Therefore, I want to find latitude θ_c and longitude φ_c. By running a linear regression on Equation (19) using points $P(\theta_b, \varphi_b)$ as data, I will get scalar values of $tan\theta_c cos\varphi_c$, $tan\theta_c sin\varphi_c$, and $\frac{cos\Omega}{cos\theta_c}$. I will also refer to these values as m_1, m_2, and b, respectively. To find φ_c, I will divide m_2 and m_1 and apply trigonometric identity (17):

$$\frac{m_2}{m_1} = \frac{tan\theta_c sin\varphi_c}{tan\theta_c cos\varphi_c} = \frac{sin\varphi_c}{cos\varphi_c} = tan\varphi_c \quad (24)$$

Equation (24) shows that I can get a value for φ_c by taking the inverse tangent of the ratio of m_2 and m_1:

$$\varphi_c = tan^{-1}\left(\frac{m_2}{m_1}\right) \tag{25}$$

To find the value of θ_c, I can substitute the calculated value of φ_c into m_1 or m_2 and solve for θ_c. In this example I will solve using m_1. First, I will divide both sides of m_1's definition in (21) by $cos\varphi_c$:

$$\frac{m_1}{cos\varphi_c} = tan\theta_c \tag{26}$$

Then, I take the inverse tangent of (26) to find θ_c:

$$\theta_c = tan^{-1}\left(\frac{m_1}{cos\varphi_c}\right) \tag{27}$$

Now, I have values for θ_c and φ_c, but I need to convert them into GPS coordinates. This is simple to do using the relationships defined in Equations (3) and (4):

$$GPS\ Latitude_{center} = 90° - \theta_c \tag{28}$$

$$GPS\ Longitude_{center} = \varphi_c \tag{29}$$

With these values found, I now know the center coordinates of the sprinkler and can express the location of the sprinkler in the field.

4.4. MATLAB Simulation

After defining my algorithm on paper, I needed a way to confirm its functionality. To do this, I implemented my algorithm into a MATLAB script that took a list of GPS points as input, ran them through my regression model, and outputted the calculated center GPS point. Once the script had been made, I chose 10 GPS points along a sprinkler's circular arc in a field in North Central Kansas. Figure 4, below, displays the simulation results using the 10 selected GPS points.

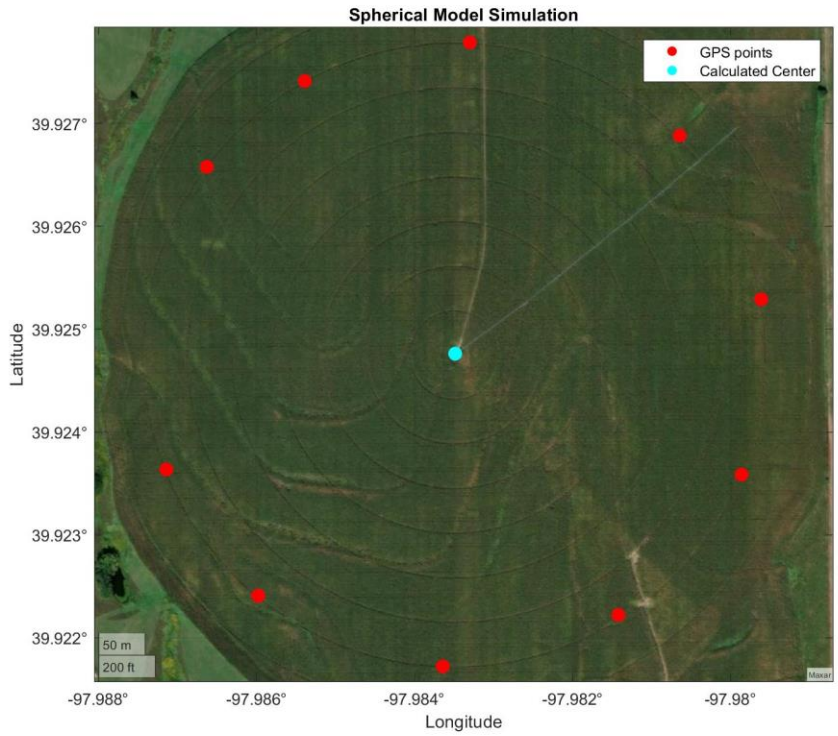


Figure 4. Simulation of my Spherical Model

The results shown above in Figure 4 indicate that my model can successfully calculate the center coordinates of a center pivot sprinkler. While I excluded any quantitative data for this example, it is clear upon visual inspection that the cyan point is in the center of the circular arc that the red points are spaced along.

5. PERFORMANCE OF CARTESIAN AND SPHERICAL METHODS

While my method using spherical coordinates clearly works, it is useful to see how its performance compares to the existing method by Trimble. In this section, I will compare computational complexity, accuracy, and the effect of field size between the two methods.

5.1. Computational Complexity

Computational complexity is analogous to an algorithm's efficiency. Since my algorithm may handle large amounts of data, it is important to consider efficiency. Typically, an algorithm's efficiency is denoted by big O notation. However, I am using built-in mathematic functions in MATLAB that make it difficult to approximate either algorithm's big O notation. Instead, I will be comparing the time it takes for each algorithm to compute versus the number of data points inputted to the algorithm. Figure 5, below, shows a graph of compute time versus number of data points.

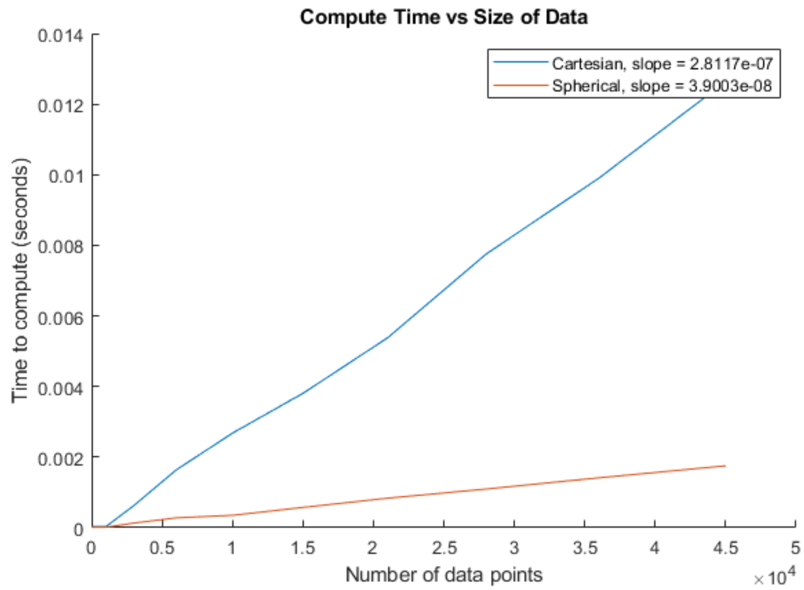


Figure 5. Graph representing relative computational complexity

By observing the results above in Figure 5, I have found that both methods appear to exhibit a linear relationship between compute time and number of data points. Since I cannot conclude either method's big O complexity, I instead fit a linear best-fit line to both curves and listed the slopes of each best-fit line in the legend of Figure 5. By calculating the ratio of these slopes $\left(\frac{2.8*10^{-7}\,{}^{sec}/_{point}}{3.9*10^{-8}\,{}^{sec}/_{point}} \approx 7\right)$, I found that Trimble's method takes roughly 7 times longer than my method to calculate the center coordinates.

5.2. Accuracy

GPS positioning is not perfect. Standalone GPS has accuracy of around 10 meters [10]. However, a more expensive GPS technology, real-time kinematic (RTK) GPS, has centimeter accuracy [11]. In this section, I will be comparing the accuracy of each method with simulated GPS points from RTK and standalone GPS.

5.2.1. Using RTK GPS

To simulate RTK GPS points, I picked 1000 coordinates directly along the circular arc of a sprinkler in a North-Central Kansas field. Since RTK GPS has centimeter accuracy, I can assume that RTK GPS points are as accurate as I can pick them. After picking the set of coordinates, I ran each model 100 times to find the average distance from the calculated center point to the known center point for each method. This average distance will be referred to as the average error. Figure 6, below, shows the results of Trimble's method and my method using these points.

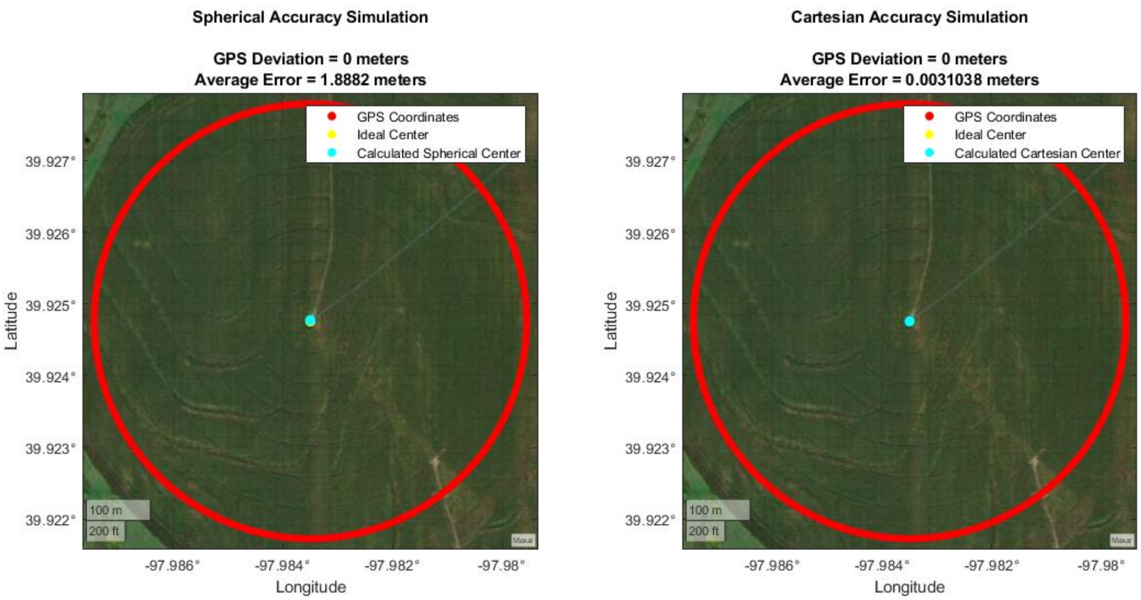


Figure 6. Simulation of RTK GPS points

As shown above in Figure 6, my method's results are displayed on the left while Trimble's method's results are on the right. The average error value is displayed above each GPS plot. Trimble's method is more accurate in this scenario than my method, having an average error of 0.003 meters while my method had an average error of 1.89 meters. However, upon visual inspection of the plot on the left, my method's calculated center (cyan) overlaps the ideal center (yellow). Thus, my method is still sufficient when using RTK GPS.

5.2.2. Using standalone GPS

To simulate coordinates found from standalone GPS, which has a 10-meter accuracy, I forced a 10-meter deviation on each of the 1000 previous RTK GPS points in a random direction. The results of the simulation are shown below in Figure 7.

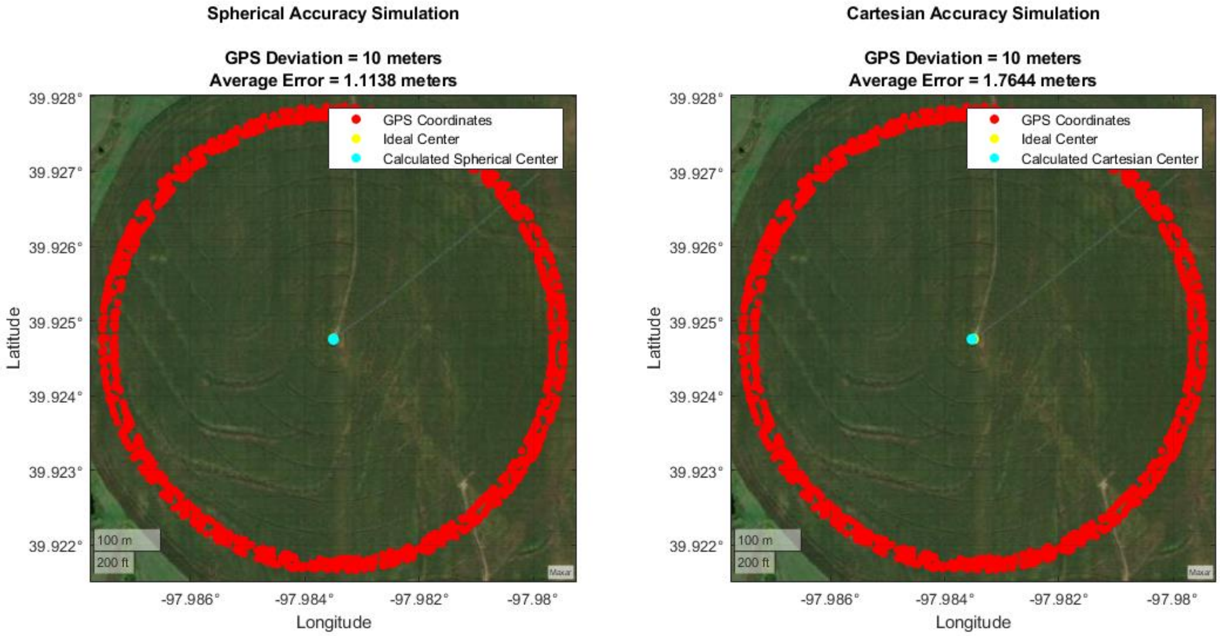


Figure 7. Simulation of 10-meter deviated GPS points

As shown above in Figure 7, my method has an average error of 1.11 meters while Trimble's method has an average error of 1.76 meters. These results are the opposite of the results for RTK GPS points in Section 5.2.1. This shows that my method can handle inaccuracies of the GPS tracker attached to the sprinkler better than Trimble's method. However, it should be noted that both models are visually sufficient for calculating the center in this scenario.

To explore the effects of GPS tracker inaccuracy further, I ran the same simulation once more, but deviated the GPS points by 100 meters instead of 10. The results of the simulation are below in Figure 8.

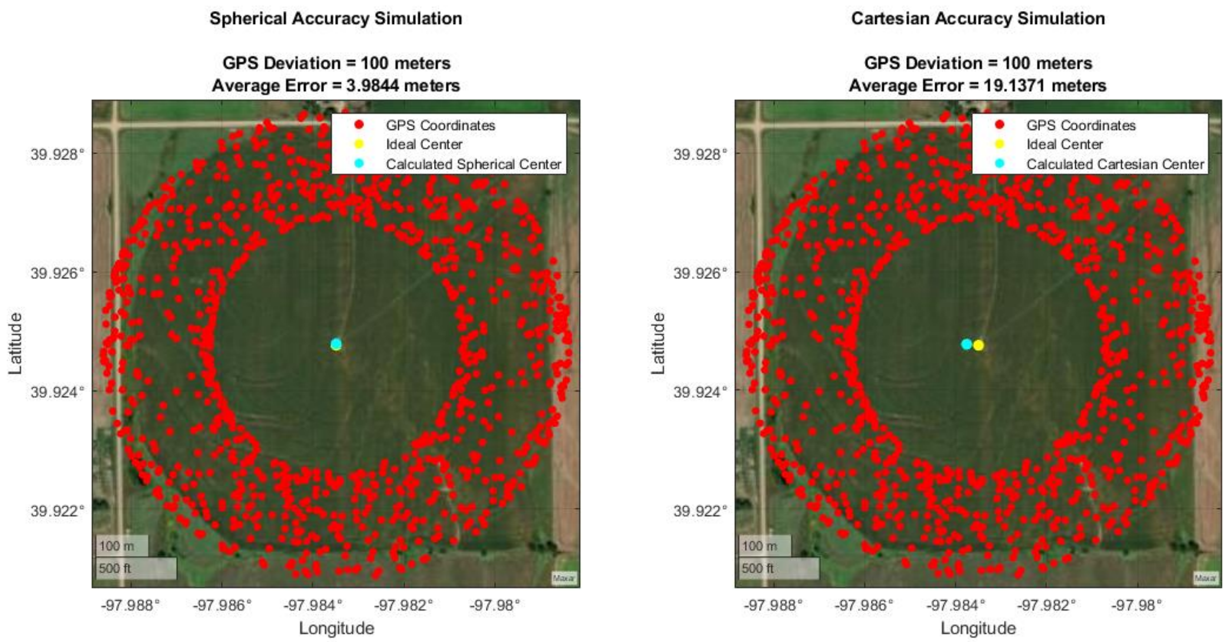


Figure 8. Simulation of 100-meter deviated GPS points

While a GPS unit should never have an accuracy of only 100-meters, the results of Figure 8 show interesting behavior. As expected, both methods had an increase in average error, but Trimble's method increased much more severely than my method. This further proves that my method can handle less accurate GPS systems better than Trimble's method.

5.3. Field Size

As I mentioned prior, Trimble's method has issues scaling with the size of the field in which it is trying to calculate the center. I ran the same simulation from Section 3.1.2 of 1000 ideal points equally spaced around a center point with a given radius for my method. Below, in Table 2, are the results.

Table 2. Effect of field size for both methods of center coordinate calculation

Radius of Field	Distance to Center Point for Cartesian Method	Distance to Center Point for Spherical Method
0.25 miles	0.01 feet	2.2 feet
1 mile	0.15 feet	0.08 feet
5 miles	3.4 feet	0.07 feet
10 miles	13.4 feet	0.4 feet
20 miles	52.9 feet	2.2 feet
100 miles	1245.6 feet	64.0 feet
200 miles	4633.0 feet	260.1 feet

The results for Trimble's method in the middle column are the same as Table 1 and the right column shows the results for my spherical method. As the size of the field increases, the accuracy of my method eventually decreases, but at a much lower rate than Trimble's method. However, Trimble's method was more accurate for 0.25-mile radius (160 acre) fields than my method. I hypothesize that this behavior is due to limitations in the precision of the coordinates when being stored in a computer. For example, two GPS points a few feet apart might only vary by 0.00001°. Therefore, for smaller distances, the accuracy of my method will be limited by the computer's precision when storing floating-point numbers. I also believe this is what caused a larger average error value in Section 5.2.1 in this report. This hypothesis is supported by my method's increased accuracy as the radius of the field increases to 1 mile and 5 miles. The computer's precision will not affect Trimble's method as much because the 2-dimensional coordinates that are used in their regression model vary by a much greater amount than the 3-dimensional GPS coordinates used in my model. Eventually, my method's accuracy begins to decrease. This is likely due to inaccuracies of the selected GPS points. I used an open-source Python library named PolyCircles [12] to create the list of GPS points for each field. While I do not know PolyCircle's method of creating the list, it does appear that the points are not perfectly centered around the given center point, thus inducing a slight amount of error for both mine and Trimble's methods for larger fields.

Most fields are much smaller than the sizes that I compared in Table 2. However, my method performs well when scaled for larger fields which allows opportunities for being used in other applications in the aviation and marine industry.

6. CONCLUSION

Overall, I have shown that my method of calculating the center coordinates of a center pivot sprinkler using spherical coordinates functions as desired. Furthermore, my method performed better than Trimble's method for every metric considered in this report.

The first metric that was measured was computational complexity. I found that both algorithms seemed to have a linear relationship between compute time and number of data points. However, Trimble's method had a linear slope 7 times greater than my method, suggesting that my method is 7 times more efficient.

The next metric that was compared was accuracy. To do this, I compared each method using ideal GPS points that represent data from an RTK GPS unit. In this scenario, I found that Trimble's method was more accurate than mine. However, when forcing a deviation to these RTK GPS points to represent data from a standalone GPS unit, my method was more accurate than Trimble's. In both cases, both methods were accurate enough to calculate a center that would be sufficient for representing the position of the sprinkler. These results are in favor of my method as RTK GPS is more expensive than standalone GPS and standalone GPS proved to be sufficient for this application.

The final metric that was discussed was field size. As discussed in Section 3.1.2, Trimble's method is best suited for smaller sized fields. I found that Trimble's method severely broke down for fields with a 100-mile radius. When the field was this size, Trimble's method found a center coordinate 1245.6 feet from the actual center while my method found a center that was only 64 feet away. However, it should be noted that for most standard-sized fields, both methods were sufficient at calculating the center.

REFERENCES

[1] Lindsay Irrigation, *FieldNET Pivot Watch "How to Install and Configure" (2020)*, (May 22, 2020). Accessed: May 01, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=1Vyb8z3autA

[2] L. J. Barker, "GPS-based control system and method for controlling mechanized irrigation systems," US6928339B2, Aug. 09, 2005 Accessed: May 01, 2022. [Online]. Available: https://patents.google.com/patent/US6928339B2/en?q=center+pivot+gps&assignee=reinke

[3] A. Samaha, "System and method for determining a pivot center and radius based on a least squares approach," US7156328B2, Jan. 02, 2007 Accessed: May 01, 2022. [Online]. Available: https://patents.google.com/patent/US7156328B2/en?q=System+and+method+for+determining+pivot+center+radius+based+on+least+squares+approach&assignee=trimble

[4] "Haversine formula - Calculate geographic distance on earth," Feb. 11, 2015. https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/ (accessed May 01, 2022).

[5] "Earth Radius by Latitude Calculator." https://rechneronline.de/earth-radius/ (accessed May 01, 2022).

[6] D. Lebedev and K. Lake, "On the influence of the cosmological constant on trajectories of light and associated measurements in Schwarzschild de Sitter space," Aug. 2013.

[7] "Polar, Spherical and Geographic Coordinates," *vvvv*. https://vvvv.org/blog/polar-spherical-and-geographic-coordinates (accessed May 01, 2022).

[8] "Spherical coordinate system," *Wikipedia*. Apr. 25, 2022. Accessed: May 01, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Spherical_coordinate_system&oldid=1084637040

[9] T. Eisenfeld and A. Karsenty, "Design and Modeling of Light Emitting Nano-Pixel Structure (LENS) for High Resolution Display (HRD) in a Visible Range," *Nanomaterials*, vol. 10, no. 2, Art. no. 2, Feb. 2020, doi: 10.3390/nano10020214.

[10] M. Saka, T. Kavzoglu, C. Ozsamli, and R. Alkan, "Sub-Metre Accuracy for Stand Alone GPS Positioning in Hydrographic Surveying," *J. Navig.*, vol. 57, pp. 135–144, Jan. 2004, doi: 10.1017/S0373463303002571.

[11] N. Chiuman, D. Atunggal, and C. A. Rokhmana, "Evaluation on the performance of NTRIP RTK positioning using multi-frequency low-cost GPS module in areas with different telecommunication signal strength," in *2019 5th International Conference on Science and Technology (ICST)*, Jul. 2019, vol. 1, pp. 1–4. doi: 10.1109/ICST47872.2019.9166367.

[12] A. Matan, *polycircles: Polycircles: WGS84 Circle approximations using polygons*. Accessed: May 01, 2022. [OS Independent]. Available: http://polycircles.rtfd.org