625.721 Final Project: Simulation of Wind Direction over the Atlantic Ocean

Matthew LeDuc

1 Introduction

The theory of statistics on convex subsets of $\mathbb{R}^n$ has been studied since at least the 7th century, when the Arab scholar Al-Khalil used permutations and combinations to list all possible Arabic words with and without vowels [5]. However, there are several areas relevant to daily life when the methods of linear statistics break down completely. For example, we know that the time halfway between 23:00 and 01:00 hours is 00:00 hours, however taking the average of 23 and 1 leads us to believe that the halfway time is 12:00 hours. We also know that the compass direction between 315 degrees (NW) and 45 degrees (NE) is not 180 degrees (S) but actually 0 deg (N), a difference of 180 degrees! Setting your compass this way would leave you hopelessly lost (ask me how I know). For this reason, over the last century the field of directional or circular statistics has been growing incredibly rapidly, starting with Richard von Mises studying the deviation of atomic weights from integer values and continuing into the present day [20]. At present, the field finds applications in modeling of temporal variations in environmental phenomena such as wind and wave direction, in the modeling of the migration patterns of birds, protein folding, and in any other area that can be assumed to be taking place on $\mathbb{S}^n$, the $n-$cylinder, or the $n-$torus. Many great examples of these sorts of processes can be found in the references, in particular in [7], [19], and [20].

Many papers have focused on the modeling of distributions of wind direction (see [1], [7], and [18], among many others, for examples). This is of particular interest to many people in industry and government, who want to know how such variations might affect, for example, the output of a wind power plant. We will continue to follow this trend in this paper. We will investigate the use of these techniques on data from the National Data Buoy Center to model time series of wind direction measured over the Atlantic Ocean.

2 Circular Statistics

Circular data can be represented as angles or as complex numbers on the unit circle. This means that a given datapoint can be represented in multiple ways, such as as an angle θ, as a unit vector $(\cos\theta, \sin\theta)^T$, or as a complex number $e^{i\theta}$. Each of these representations is useful for different applications and each will be used in this work. In addition, there are multiple ways to represent the values of the angles themselves and multiple ways to select the 0 point and positive direction of rotation. However, in spite of these possibilities we want two people who are obsesrving the same data but using different conventions to be able to get the same results. This invariance to coordinates is the main issue that separates ordinary statistics, where it is almost never worried about, and circular statistics. In this section we will discuss the definitions of common statistical quantities on the circle, which will allow us to analyze data and do comparisons in a similar way to ordinary statistics.

2.1 The Mean

Consider a set of unit vectors $x_1, ..., x_n$ with corresponding angles $\theta_1, ..., \theta_n$. One intuitive way to find the mean is by setting $z_k = e^{i\theta_k}$, in which case $\bar{z} = \frac{1}{n}\sum_{k=1}^{n} z_k$. Then the *mean resultant length* $R = \|\bar{z}\|$ and the *mean direction* is given by $\bar{\theta} = \angle\bar{z}$. The mean $\bar{z}$ represents the location of the center of mass of the datapoints. If we chose not to convert to complex numbers, we could find the angular mean by bearing in mind its relationship with the center of mass of the datapoints. The mean direction is the direction of the sum $x_1 + ... + x_n$. Since the coordinates of each x_k are given by $(\cos\theta_k, \sin\theta_k)^T$, the coordinates of the center of mass of the datapoints $(\bar{C}, \bar{S})^T$ are given by

$$\begin{aligned} \bar{C} &= \frac{1}{n}\sum_{k=1}^{n} \cos\theta_k \\ \bar{S} &= \frac{1}{n}\sum_{k=1}^{n} \sin\theta_k \end{aligned} \tag{2.1}$$

and R and $\bar{\theta}$ are the solutions to the system of equations

$$\begin{aligned} \bar{C} &= R\cos\bar{\theta} \\ \bar{S} &= R\sin\bar{\theta} \end{aligned} \tag{2.2}$$

assuming $R > 0$. Then the value of R can be assumed from trigonometry to be $R = \sqrt{\bar{C}^2 + \bar{S}^2}$, and the value of $\bar{\theta}$ can be found using the four-quadrant arctangent function [19]. The mean resultant length R is between 0 and 1, being close to 0 when the data is spread over the circle and close to 1 when it is highly concentrated. Thus, it can be thought of as a measure of the concentration of a dataset. However, we should note that the data set made by $\theta_1, ..., \theta_n, \theta_1 + \pi, ..., \theta_n + \pi$ has $R = 0$, so $R = 0$ does not imply that the data is spread evenly over the circle, only that the datapoints are spread symmetrically.

2.2 Variance, Dispersion, and Standard Deviation

For most applications, R is perfectly adequate as a measure of spread [19], however it is sometimes necessary to consider other measures for comparison with ordinary data. The *circular variance* is defined as

$$V = 1 - R \tag{2.3}$$

and since $0 \leq R \leq 1$, it follows that $0 \leq V \leq 1$ as well.

We may also wish to consider the dispersion of a set of angles. We do this by considering a useful measure of angular distance: the cosine distance given by

$$\delta = 1 - \cos(\theta_1 - \theta_2) \tag{2.4}$$

This naturally suggests one way of measuring dispersion of a set of angles about a given angle α:

$$D(\alpha) = \frac{1}{n}\sum_{k=1}^{n}(1 - \cos(\theta_k - \alpha)) \tag{2.5}$$

It can be shown that equation 2.5 has a minimum of V at $\alpha = \bar{\theta}$, which is analogous to the result on the line that $\frac{1}{n}\sum_{k=1}^{n}(x_k - u)^2$ takes on a minimum at $u = \bar{x}$, and that value is a biased estimator of the sample variance [19].

We can derive an analogue for the standard deviation by transforming the variance V. One appropriate statistic is the sample *circular standard deviation* given by

$$v = \sqrt{-2\log(1-V)} = \sqrt{-2\log(R)} \tag{2.6}$$

which takes values on $\mathbb{R}_{\geq 0}$ instead of $[0, 1]$. For small values of V this takes the form

$$v \approx \sqrt{2(1-R)} \tag{2.7}$$

The easy analogue to ordinary statistics leads some authors to refer to $2(1-R)$ as the circular variance instead of equation 2.3 [19].

2.3 Trigonometric moments

Recall from equation 2.1 that $\bar{C}$ and $\bar{S}$ are defined as the average cos/sin respectively of the angles θ_k. As alluded to in section 2.1, these are related to the mean resultant length and the mean direction of the x_k. They can be combined to form the *first trigonometric* *mement about zero* given by

$$m_1' = \bar{C} + i\bar{S} = Re^{i\bar{\theta}} \tag{2.8}$$

This can be extended to define the p^{th} *trigonometric moment about zero* for $p = 1, 2, 3, ...$ by writing

$$\begin{aligned} m_p' &= a_p + ib_p \\ a_p &= \frac{1}{n}\sum_{k=1}^{n}\cos(p\theta_k) \\ b_p &= \frac{1}{n}\sum_{k=1}^{n}\sin(p\theta_k) \end{aligned} \tag{2.9}$$

Then, one can see from euler's formula that $m_p = R_p e^{i\bar{\theta_p}}$, where R_p and $\bar{\theta_p}$ are the mean resultant length and direction of the data $p\theta_1, ..., p\theta_n$. The p^{th} moment about the mean m_p is calculated in the same way, however the argument of the trigonometric function is $p(\theta_k - \bar{\theta})$ instead. It can be shown that $m_1 = R$. The population moments will be introduced in our discussion of distributions on the circle.

Now that we have a handle on common summary statistics on the circle, we can move to a discussion of common distribution functions on the circle, and then to models for data on the circle.

3 Distributions on the Circle

Now that we have seen the basics of circular statistics, we will introduce the most important circular probability distributions.

3.1 The circular uniform distribution

The easiest generalization from the real line to the circle is the uniform distribution. This is given by the probability density

$$f_X(x) = \frac{1}{2\pi} \tag{3.1}$$

which has the associated distribution function

$$F_X(x) = \frac{x}{2\pi} \tag{3.2}$$

This has all the same properties as the uniform distribution of the real line, so we will not spend time discussing it in detail.

3.2 The von Mises Distribution

There are several different distributions that can be used for modeling wind direction data, however the most common is the von Mises distribution [1]. Its probability density function is given by

$$f_X(x) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{3.3}$$

and its cumulative density function is given by [19]

$$F_X(x) = \frac{1}{I_0(\kappa)} \int_0^x e^{\kappa \cos(x-\mu)} dx \tag{3.4}$$

where μ is the angular mean, $\kappa \in [0, \infty)$ is a parameter that determines the spread of the distribtuion, and $I_0(\kappa)$ is the zeroth order modified Bessel function of the first kind given by

$$I_r(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} \cos(r\theta) e^{\kappa \cos(\theta)} d\theta, r = 0, \pm 1, \pm 2, ... \tag{3.5}$$

We will denote the von Mises distribution with parameters μ and κ by $M(\mu, \kappa)$. The population mean direction is given by μ. The n^{th} moment of $z = e^{i\theta}$ is given by (as one might expect from linear statistics)

$$m_n = \int_0^{2\pi} \frac{z^n}{I_0(\kappa)} e^{\kappa \cos(\theta-\mu)} d\theta \tag{3.6}$$

where $\|m_1\| = R$. Knowing that $z^n = \cos(n\theta) + i \sin(n\theta)$ and referring to equation 3.5 we can see that

$$R = \frac{I_1(\kappa)}{I_0(\kappa)} \tag{3.7}$$

This is a potentially difficult expression to evaluate, however luckily we can leverage the asymptotic expansions of $I_r(\kappa)$ for small and large κ to find that in these cases

$$R = \frac{\kappa}{2}\{1 - \frac{1}{8}\kappa^2 + \frac{1}{48}\kappa^4 - \frac{11}{3072}\kappa^6 + ...\} \tag{3.8}$$

for small values of κ and

$$R = 1 - \frac{1}{2\kappa} - \frac{1}{8\kappa^2} - \frac{1}{8\kappa^3} + o(\kappa^{-3}) \tag{3.9}$$

for large values of κ [19].

3.2.1 Maximum Likelihood Estimation

We will determine the approximate distribution of the wind direction via maximum likelihood estimation. Consider a set of n data points $x_1, x_1, ..., x_n$that are drawn from some distribution $f_X(x;\theta)$ where θ is an unknown parameter or vector of parameters. We can estimate θ by means of maximizing the likelihood function of the distribution. The likelihood function for θ is given by

$$L(\theta) = \prod_{k=1}^{n} f_X(x_k|\theta) \tag{3.10}$$

and the maximum likelihood estimate is the value $\hat{\theta}$ such that $L(\hat{\theta}) \geq L(\theta)$ for all values of θ and is the solution of the equation $\nabla L(\theta) = 0$, or equivalently $\nabla \log(L(\theta)) = 0$ (called the log-likelihood) which can often reduce the difficulty of the computation [17]. The log-likelihood function for the von Mises distribution given n data points is given by

$$\log(L(\kappa, \mu)) = -n \log(2\pi I_0(\kappa)) + \sum_{k=1}^{n} e^{\kappa \cos(x_k - \mu)} \tag{3.11}$$

Taking the appropriate gradient and setting it to zero, we can derive the maximum likelihood estimators $\hat{\mu}$ and $\hat{\kappa}$. Let the data points $x_1, x_2, ..., x_n$ be drawn from the interval $[0, 2\pi)$ according to some distribution and consider the transformation $z_k = e^{ix_k}$. Then the smaple mean is given by $\bar{z} = \frac{1}{n}\sum_{k=1}^{n} z_k$ and the maximum likelihood estimate for μ is given by $\hat{\mu} = \angle \bar{z}$. The estimator for κ is given by [4]

$$\bar{R} = \frac{I_1(\hat{\kappa})}{I_0(\hat{\kappa})} \tag{3.12}$$

where $\bar{R} = \|\bar{z}\|$. Note that this is the expression in equation 3.7 with the sample values of R and κ replacing the population values. This equation can be solved via your favorite rootfinding algorithm. The author's favorite rootfinding method is the Newton-Rhapson algorithm so it is employed in this paper.

4 NDBC Data

Our dataset of interest is the NOAA National Data Buoy Center wind direction data [22]. The NDBC operates a network of buoys and coastal stations throughout the world that provide important oceanograhic and atmospheric data such as wind speed and direction, significant wave height, sea surface temperature, and barometric pressure. These observations are used by federal, state, academic, and private stakeholders for weather forecasting, developing oceanic and atmospheric models, environmental monitoring, and many other purposes [24]. For the purposes of this work, we will focus on data from buoys 41010 and 41047 collected in 2019 and 2020. The buoys are both in the Atlantic Ocean near the United States, with buoy 41010 on the Blake Plateau and buoy 41047 near the Hatteras Plain (see figure 4.1).

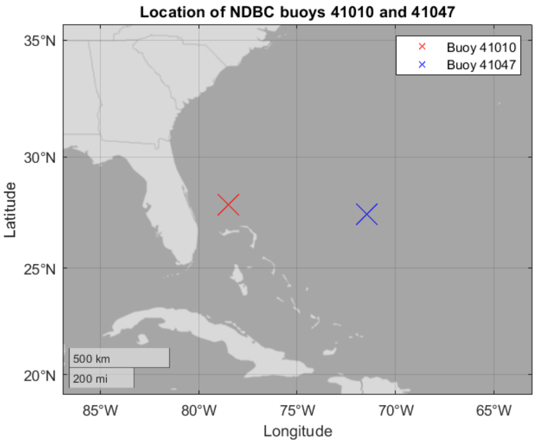
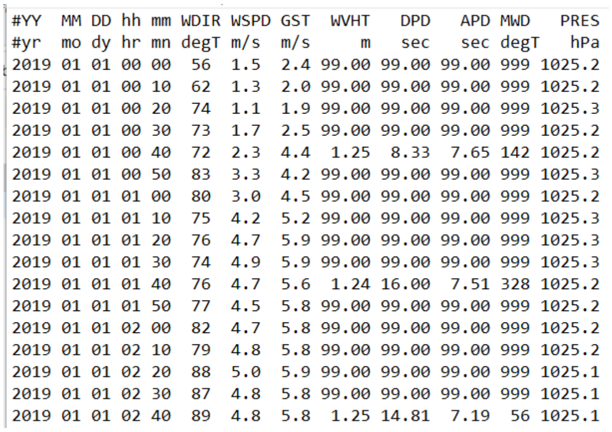


Figure 4.1: Location of NDBC buoys 41010 (red) and 41047 (blue)

The buoy anemometers are approximately 4 m above sea level, and wind direction data is reported every 10 minutes as integer angles in the interval $[0, 360)$ measured clockwise from north. The data for each year is written into a text file, with each line representing one 10 minute time period (shown in figure 4.2, wind direction data is in the column titled WDIR). For our purposes, we will split the data up by month in an attempt to minimize seasonal affects. Missing values are represented by values of either 99.0 or 999.0 depending on the data column, which makes them easy to filter out. When there are only a few isolated missing data points, they are replaced by taking the angular average of the data points on either side. If they are not isolated, the month of data is not processed. This is not usually an issue that we come across in this dataset. Once the data has been read in, it is converted from angles in $[0, 360)$ to $[-\pi, \pi)$ for processing. This data is matched to a von Mises distribution by means of maximum likelihood estimation for the purposes of modeling.



```
#YY  MM DD hh mm WDIR WSPD GST  WVHT   DPD   APD MWD   PRES
#yr  mo dy hr mn degT m/s  m/s     m   sec   sec degT   hPa
2019 01 01 00 00  56  1.5  2.4 99.00 99.00 99.00 999 1025.2
2019 01 01 00 10  62  1.3  2.0 99.00 99.00 99.00 999 1025.2
2019 01 01 00 20  74  1.1  1.9 99.00 99.00 99.00 999 1025.3
2019 01 01 00 30  73  1.7  2.5 99.00 99.00 99.00 999 1025.2
2019 01 01 00 40  72  2.3  4.4  1.25  8.33  7.65 142 1025.2
2019 01 01 00 50  83  3.3  4.2 99.00 99.00 99.00 999 1025.3
2019 01 01 01 00  80  3.0  4.5 99.00 99.00 99.00 999 1025.2
2019 01 01 01 10  75  4.2  5.2 99.00 99.00 99.00 999 1025.3
2019 01 01 01 20  76  4.7  5.9 99.00 99.00 99.00 999 1025.3
2019 01 01 01 30  74  4.9  5.9 99.00 99.00 99.00 999 1025.3
2019 01 01 01 40  76  4.7  5.6  1.24 16.00  7.51 328 1025.2
2019 01 01 01 50  77  4.5  5.8 99.00 99.00 99.00 999 1025.2
2019 01 01 02 00  82  4.7  5.8 99.00 99.00 99.00 999 1025.2
2019 01 01 02 10  79  4.8  5.8 99.00 99.00 99.00 999 1025.2
2019 01 01 02 20  88  5.0  5.9 99.00 99.00 99.00 999 1025.1
2019 01 01 02 30  87  4.8  5.8 99.00 99.00 99.00 999 1025.1
2019 01 01 02 40  89  4.8  5.8  1.25 14.81  7.19  56 1025.1
```

Figure 4.2: Screenshot of raw data for NDBC buoy 41047

5 Modeling

Over the years, a variety of models have been introduced for generating data with a proper directional distribution. In this section, we will explore two common ways of generating circular data: The binding density method for generating Markov processes on the circle and the von Mises process, which is a circular analogue of the Ornstein-Uhlenbeck process. We will first discuss autocorrelation and its definition for circular data, as in order ot generate realistic data we need to be able to generate data with a substantial autocorrelation.

5.1 Correlation of circular data

The autocorrelation of a stochastic process is the Pearson correlation of values of the process separated by a given length of time. If the process $\{X_t\}$ has zero mean and time independent variance, then the autocorrelation function is given by

$$\rho_X(\tau) = E\left[X_t \bar{X}_{t+\tau}\right] \tag{5.1}$$

where $\bar{X}$ denotes the complex conjugate. Because of the fact that the mean and variance are time independent, the autocorrelation only depends on the distance betrween samples and therefore $\rho_X(\tau)$ is an even function of the lag τ [10]. It also achieves a maximum value of $E\left[X_t^2\right] = \mathrm{Var}(X_t)$ for zero-mean processes. However, this definition of the autocorrelation is used for data drawn from the real line, and our data is drawn from the unit circle. Therefore, this definition of the autocorrelation may not be appropriate for use in our situation.

As we can readily see from any selection of data (such as what is presented in figure 5.1), real wind direction data is highly autocorrelated. There are several competing definitions of the sample autocorrelation, however we will follow the definition of the sample correlogram used in [8]. This will allow us to select an appropriate model and parameters for our data generation by means of matching the sample correlogram.

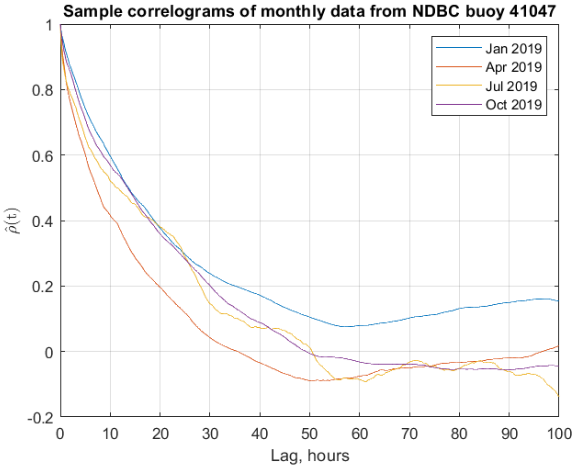


Figure 5.1: Autocorrelation of NDBC buoy data from buoy 41047

The sample correlogram $\hat{\rho}(\tau)$ as defined in [8] is given by

$$\hat{\rho}(\tau) = \frac{\det\left(\sum_{k=1}^{n-\tau} x_k x_{k+\tau}^T\right)}{\sqrt{\det\left(\sum_{k=1}^{n-\tau} x_k x_k^T\right)\det\left(\sum_{k=\tau+1}^{n} x_k x_{k+\tau}^T\right)}} \tag{5.2}$$

where $x_k = (\cos(\theta_k), \sin(\theta_k))^T$ and τ is the lag in samples. This allows us to determine the autocorrelation spectrum of our data, and therefore compare the data to what we generate with our model. An example of several autocorrelation spectra of NDBC data are shown in figure 5.1. Note that for values of $\hat{\rho} > 0.2$ the correlograms have the shape of an exponential decay. We will come back to this figure and leverage this fact later when discussing fitting our model output to data.

5.2 Markov Processes on the Circle

In [13] , Johnson and Wehrle developed an algorithm for generating stationary Markov processes on the circle. Consider three distributions on the circle given by $f_1(\theta), f_2(\phi), g(\cdot)$ and let $F_k(\theta)$ be the distribution function of f_k. The distribution $g(\cdot)$ is called the binding density. We wish to find a joint distribution function $f(\theta, \phi)$ such that the marginal densities are f_1 and f_2. Johnson and Wehrle showed that

$$f_{\Theta,\Phi}(\theta, \phi) = 2\pi g\{2\pi(F_\Theta(\theta) - F_\Phi(\phi))\} f_\Theta(\theta) f_\Phi(\phi) \tag{5.3}$$

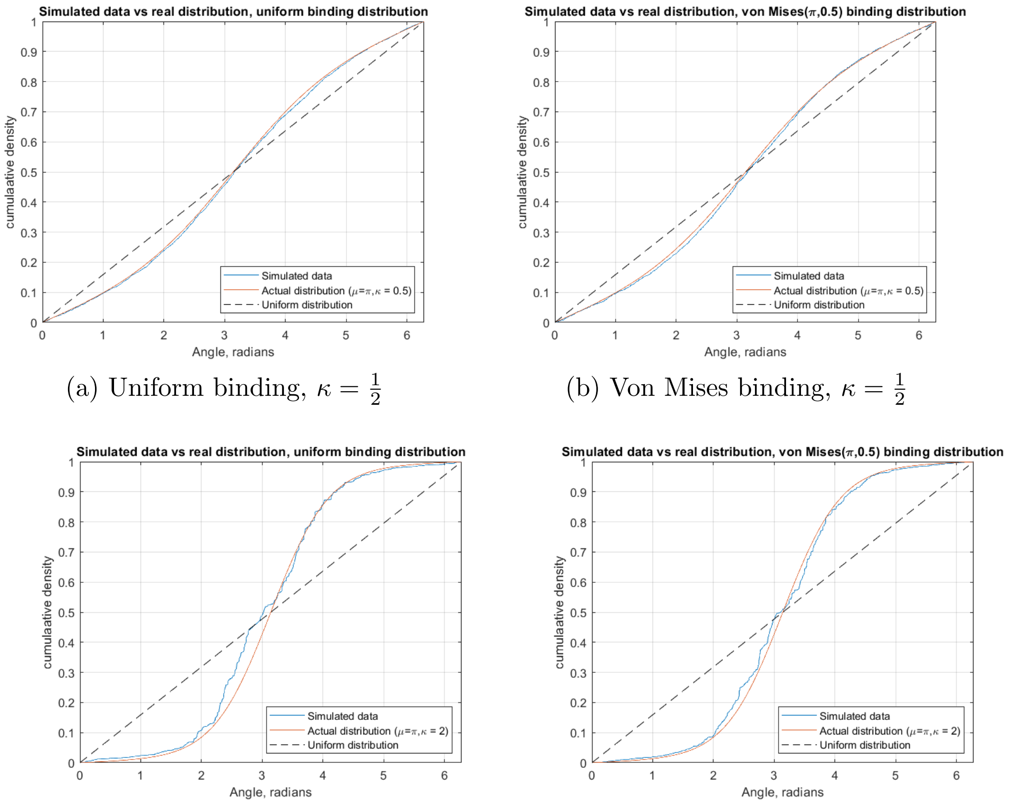
satisfies this condition. This can be seen by substituting $z = F_k(x)$ and integrating to calculate the marginals. They went on to define a Markov process on the unit circle by choosing $f = f_1 = f_2$ and setting the initial density $p(\theta_0) = f(\theta_0)$. We can clearly see from equation 5.3 that the conditional density of each θ_t is

$$p(\theta_t|\theta_{t-1}, \theta_{t-2}, ..., \theta_0) = p(\theta_t|\theta_{t-1}) = 2\pi g\{2\pi(F(\theta_t) - F(\theta_{t-1}))\}f(\theta_t) \tag{5.4}$$

since each θ_t has the same marginal distribution.

We can use this to apply the Metropolis-Hastings algorithm [21] to generate data that is distributed as f. Let $\{\Theta_t\}$ be a sequence of random variables on the set $I = [0, 2\pi)$ with target distribution $f(\theta)$. First, choose some value $y \in I$ from the transitional density $p(y|\theta_t)$ given in equation 5.4. Now, let

$$\alpha = \min\{1, \frac{f(y)p(y|\theta_t)}{f(\theta_t)p(\theta_t|y)}\} \tag{5.5}$$



(a) Uniform binding, $\kappa = \frac{1}{2}$

(b) Von Mises binding, $\kappa = \frac{1}{2}$

(c) Uniform binding, $\kappa = 2$

(d) Von Mises binding, $\kappa = 2$

Figure 5.2: Simulated data vs target distribution for various distributions and binding densities, $n = 4000$

Then, set $\theta_{t+1} = y$ with probability α and $\theta_{t+1} = \theta_t$ with probability $1 - \alpha$, and step forward. The expression in α can be thought of as a sort of likelihood function, where values of y that are more "likely" to have come from $f(\theta)$ than θ_t get accepted automatically and others only get accepted with some probability $\alpha < 1$ depending on their similarity to the expected distribution. Results for $\theta \in M(\mu = \pi, \kappa = \{\frac{1}{2}, 2\})$ with $n = 4000$ and both a uniform and $M(\pi, \frac{1}{2})$ binding density are shown in figure 5.2. Based purely on these results it appears that for larger values of κ the distribution of $\{\Theta_t\}$ converges to the target distribution more slowly.

This is a nice result, however in order to simulate realistic data our model must also have realistic autocorrelation. As we can see in figure 5.3, the simulated data has almost no autocorrelation, but data from NDBC buoys is highly autocorrelated, as we have seen for example in figure 5.1. In spite of what appears to be quick convergence to the stationary distribution, this lack of autocorrelation means that we cannot use this to generate time series of realistic data.

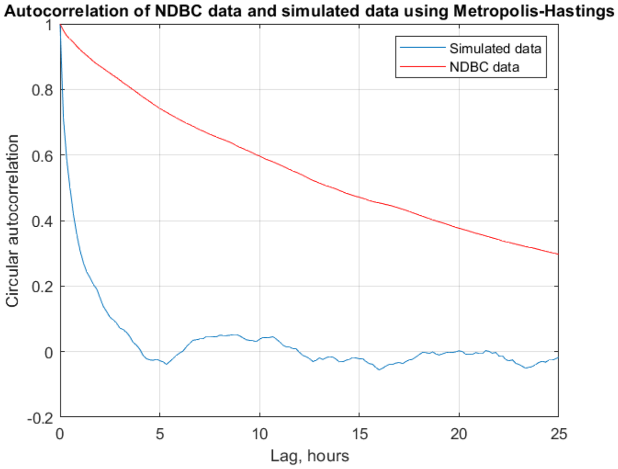


Figure 5.3: Autocorrelation of data generating using Metropolis-Hastings vs real data

5.3 The von Mises Process

There are a variety of autoregressive models for data on the circle of varying complexity (see [6], [8], [11], or [14] among others, for an overview of many commonly used ones), however we will use a model based off the well-understood Ornstein-Uhlenback process [23]. The Ornstein-Uhlenbeck process is the solution to the stochastic differential equation

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \tag{5.6}$$

where dW_t is a Wiener process, that is, a continuous time stochastic process with independent, normally distributed increments with mean 0 and variance equal to the time step so that $W_{t+\Delta t} - W_t \in N(0, \Delta t)$. This process is a modification of the continuous random walk, and can be thought of as a continuous analogue to the AR(1) process [23].

One of the first continuous time processes on the circle was proposed in [15]. This process, known as the von Mises process, can be thought of as a continuous time analogue to the Ornstein-Uhlenbeck process [9]. It was defined as the solution to the familiar looking stochastic differential equation

$$d\Theta_t = \alpha \sin(\mu - \Theta_t)dt + \sigma dW_t \tag{5.7}$$

where again W_t is a Wiener process, $\alpha > 0$ is a drift coefficient, μ is the mean direction, and σ is a diffusion coefficient. This process has a unique stationary distribution given by the $M(\mu, \frac{2\alpha}{\sigma^2})$ distribution. By assuming unit time steps, this can be easily reformulated into a stochastic difference equation as

$$\theta_{t+1} - \theta_t = \alpha \sin(\mu - \theta_t) + \sigma\epsilon_t \tag{5.8}$$

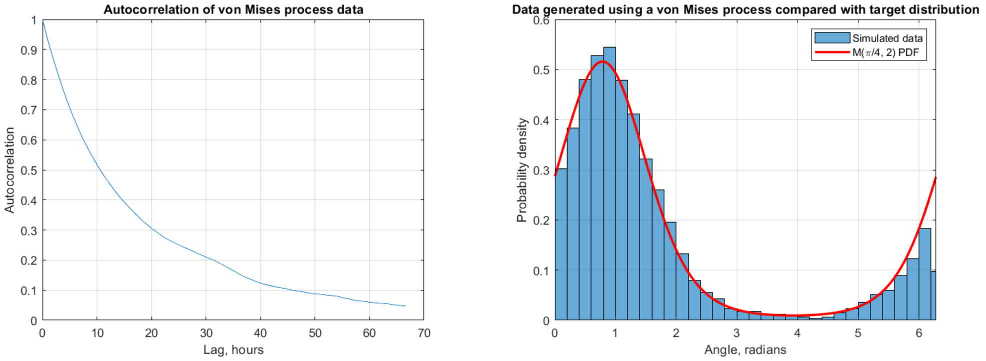
where $\epsilon \in N(0, 1)$. If we know our stationary distribution, we can develop a model using equation 5.8 to generate autocorrelated data with the desired distribution. This formulation also allows us to leverage regression techniques to determine the value of α appropriate to match our data.

An example of the data generated by this model is shown in figure 5.4. The data shown consists of 50000 samples generated with a target stationary distribution given by the $M(\pi/4, 2)$ distribution and with $\alpha = 0.01$. The data is strongly autocorrelated with a $1/e$ decorrelation time of approximately 15 hours assuming a sample rate of 6 samples per hour (which aligns with the NDBC data as seen in figure 5.1). Due to its inherent and tractable autocorrelation, this will be our model of choice through the remainder of the paper.

6 Results

Using the material covered thus far, a MATLAB model was developed based on equation 5.8 to generate realistic time series of wind directions based on the NDBC data. The code was written specifically for this, however there are several software pacakges meant for handling circular data that could have been leveraged. One commonly used toolbox is the MATLAB CircStat toolbox described in [3].

Our first task is determining appropriate values of the parameter α in equation 5.8.



(a) Sample autocorrelation of data generated from a von Mises process

(b) Simulated data vs stationary distribution

Figure 5.4: Autocorrelation of simulated data and comparison with stationary distribution

Our first attempt to match this parameter will leverage linear regression. By setting $\Delta\theta_t = \theta_{t+1} - \theta_t$ we can write equation 5.8 in a format amenable to linear regression as

$$\vec{\Delta\theta} = \begin{bmatrix} \Delta\theta_1 \\ \vdots \\ \Delta\theta_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\mu - \theta_1) \\ \vdots & \vdots \\ 1 & \sin(\mu - \theta_{n-1}) \end{bmatrix} \begin{bmatrix} c \\ \alpha \end{bmatrix} + \begin{bmatrix} \sigma\epsilon_1 \\ \vdots \\ \sigma\epsilon_{n-1} \end{bmatrix} = \beta\vec{x} + \sigma\vec{\epsilon} \tag{6.1}$$

Assuming that the noise term is relatively small, we can write an expression for α as

$$\begin{bmatrix} c \\ \alpha \end{bmatrix} = (\beta^T\beta)^{-1}\beta^T\vec{\Delta\theta} \tag{6.2}$$

Due to the structure of equation 5.8, we expect that c ought to be nearly zero. However, results from this method are mixed. As seen in figure 6.3, agreement between the data and model autocorrelation functions is mixed. While in some cases, as for month 4, the autocorrelation spectra are well-matched for moderate time lags, for others such as month 1 they are not even nearly similar past very short lags[1].

However, as we can recall from the discussion in section 5.1 and see in figures 5.1 and 6.3, the autocorrelation is, at least where $\hat{\rho} \geq 0.2$, very nearly an exponential decay. With this in mind, we can suggest a different way to determine an appropriate value of α by matching the $1/e$ decorrelation time. If we want the data to be exponentially autocorrelated with a $1/e$ decorrelation time of n samples, we can set $\alpha = \frac{1}{n}$, and then we will have an autocorrelation function $\rho(m) = e^{-\frac{m}{n}}$, where m is in samples. By setting n to the $1/e$ decorrelation of the data in samples and setting α accordingly, we generate the results shown in figure 6.2.

[1] We do not care about matching long-time (i.e. $\geq$ 30 hours) autocorrelation since the data is not meaningfully correlated that far out, but we do care for short times where data is highly autocorrelated

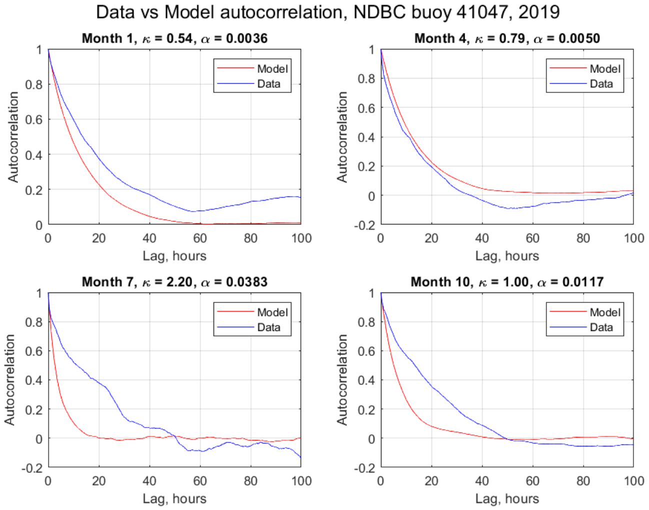


Figure 6.1: Model vs data autocorrelation with α calculated by linear regression

We can see that the results match the data better in month 7 and month 10, however the performance is worse for months 1 and 4. In this method, it appears that we are better able to match the autocorrelation for high values of κ than lower values.

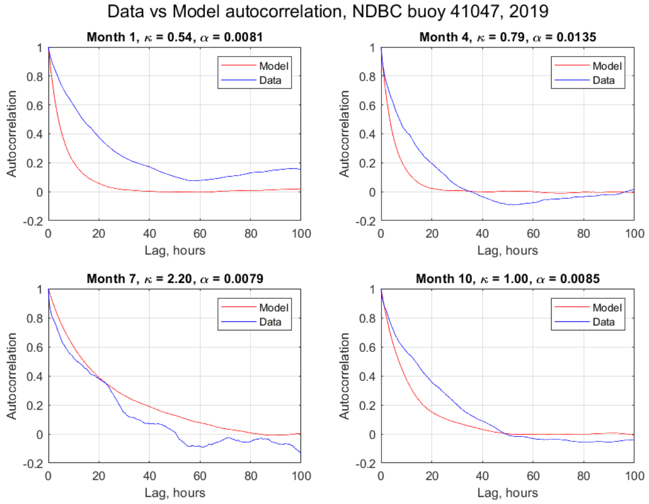


Figure 6.2: Model vs data autocorrelation with α calculated by matching decorrelation time

We can see this trend continue with data from buoy 41010 taken in 2018, where the months with higher values of κ have a modeled autocorrelation that seems to provide a decent fit to the data autocorrelation. Generally, this method performs best when $\kappa > 1$.

With this in mind, we want to determine what the actual effect of κ on the error in decorrelation time might be for both of our methods of calculating α: by matching decorrelation time and by performing linear regression. To do the calculation, every usable month

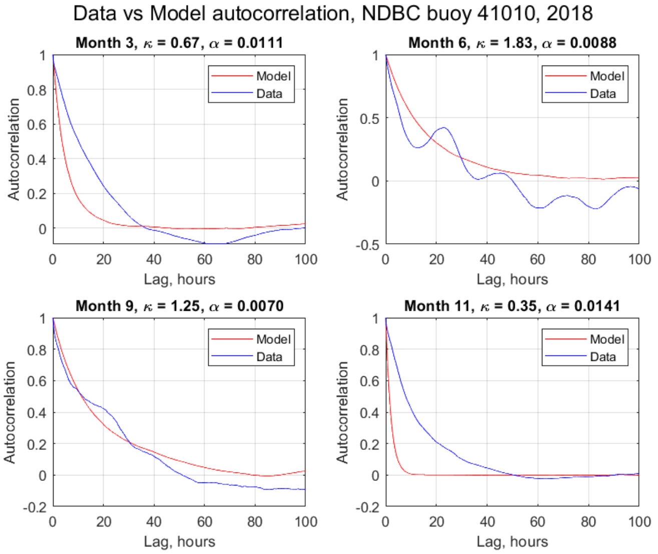
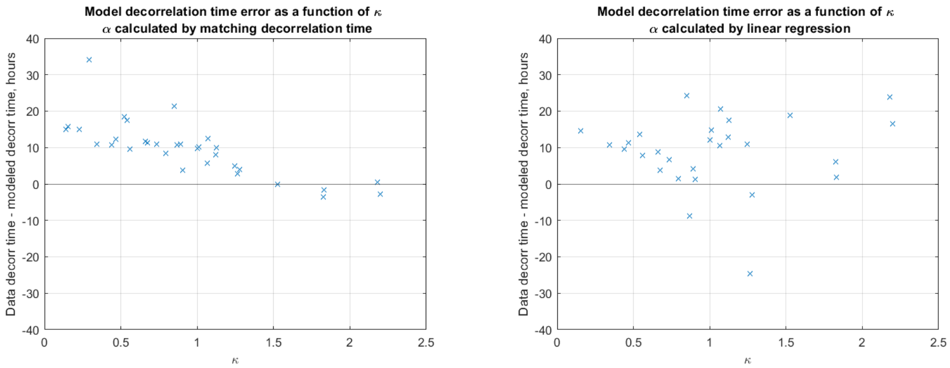


Figure 6.3: Buoy 41010, autocorrelation generated by matching decorrelation time

of data available was processed, and a maximum likelihood estimate of κ was calculated. Then the value of α was estimated by linear regression and by matching decorrelation time, and the difference between the decorrelation time shown by the data and the model in hours was recorded. By doing this, we see hard evidence for what was postulated in the previous paragraph: As the value of κ increases, we are better able to match the decorrelation time seen in the data when we set α to match it, but when we use linear regression to find α we are less able to properly match the decorrelation time. By choosing α, we are able to match decorrelation time to within about 5 hours for $\kappa \geq 1.5$ and nearly within 10 hours for $\kappa \geq 1$, which, while far from perfect, is notably better than the results when α is calculated via linear regression.

This is not entirely unexpected, since if we expect equation 5.8 to be a good match for the data we can see that the variance of the noise term σ^2 is related to α by the relation $\kappa = \frac{2\alpha}{\sigma^2}$, so $\sigma^2 = \frac{2\alpha}{\kappa}$ and the noise term is similar in size to the magnitude of the $\alpha \sin(\mu - \theta_t)$ term or even much larger than it, which makes linear regression difficult. And when κ is large, the data is more concentrated around μ so $\alpha \sin(\mu - \theta_t) \approx \alpha(\mu - \theta_t)$ more often, so our von Mises process is more like an $AR(1)$ process and the autocorrelation is closer to an exponential decay, which we assumed to calculate α in the first place.

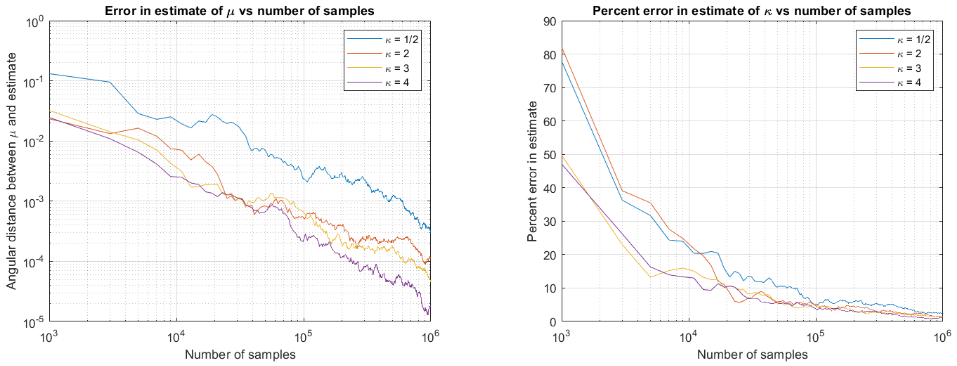
Lastly, we want to be able to demonstrate that we generate data that matches the assumed distributions. While we have seen an example of this in figure 5.4 for a large number of datapoints, we want to get an idea of how quickly we converge to the assumed distribution. noting that there are only about 4000 samples of wind direction taken in each



(a) α chosen to match decorrelation (b) α chosen by linear regression

Figure 6.4: The effect of κ on matching autocorrelation for both methods of estimating α

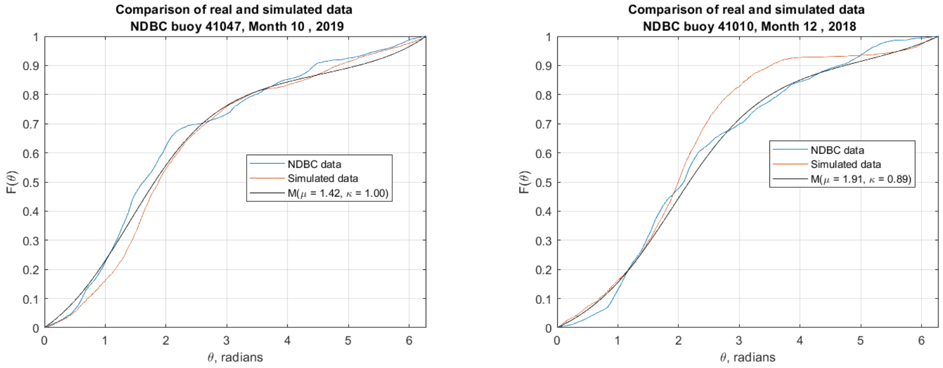
month, we would like to be able to converge to the stationary distribution without exceeding that value too much. To investigate convergence, several sequences of 10^6 datapoints were generated with $\mu = 0$ and values of κ ranging from 0.5 to 4 to cover the approximate range of values of κ seen in the data. Maximum likelihood estimates of μ and κ were calculated for the first n samples, for $10^3 \leq n \leq 10^6$. Error in μ is measured by equation 2.4 and error in κ is measured by percent error. From this, we can see that while the model converges to the mean of the stationary distribution quickly, it takes much longer to get an estimate of κ that is within 10% of the stationary value, in this case requiring well over 10^4 samples. More importantly, for 4000 samples, the error in the estimate of κ can be more than 20%. The convergence is much slower than for the data generated using the Metropolis-Hastings method, which generated data matching the target distribution within the desired 4000 points.



(a) Angular error in the estimate of μ vs n (b) Percent error in the estimate of κ vs n

Figure 6.5: Error in the estimates of μ and κ for varying κ

Whether this is an acceptable level of error or not depends on how exact you wish your modeling to be. In our case, this slow convergence to the stationary distribution means that unless we are generating large amounts of data we cannot guarantee that the data approximates the stationary distribution well. This can be readily seen by plotting the empircal CDFs of the NDBC data and the same number of simulated data points on the same axes. While sometimes we are able to get data that has a reasonably similar distribution to the NDBC data, we are just as often unable to do so, as seen in figure 6.6. Sometimes we get close to the assumed stationary distribution (shown in black) as when mimicking the buoy 41047 data from October 2019 (or, at least, we are no further away than the data is), and sometimes we are not remotely close, as seen when we try to simulate the data from buoy 41010 in December 2018. Someone with more writing space would use a test of distributional similarity on the circle such as the Kuiper Test described in [16] to show statistics and give a sense for how this changes with increased sample size but that would put me well over the page limit.

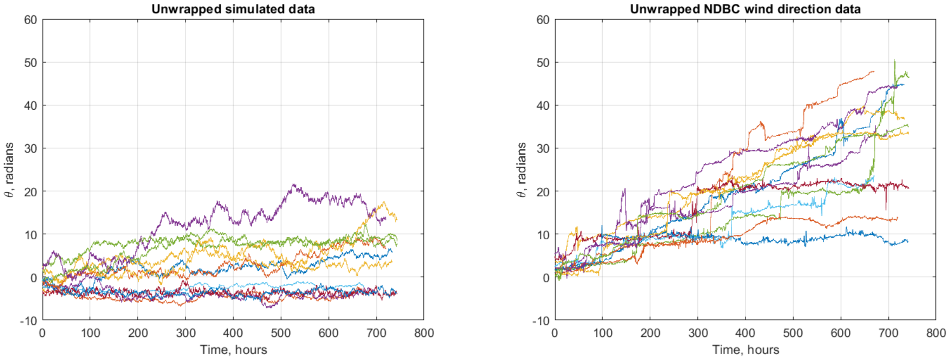


(a) Buoy 41047, Oct 2019

(b) Buoy 41010, Dec 2018

Figure 6.6: Comparison between simulated and real NDBC data

So, while the model is able to generate autocorrelated data with a von Mises distribution, the data does not match the spread of the stationary distribution of the von Mises process well without generating a large amount of data points and the autocorrelation is difficult to get exactly right. There is one more feature of the data thar the model does not properly capture. When the data is unwrapped, it appears to have a strong drift term. This suggests that the wind direction is varying in a periodic way, and while it may be concentrated in a certain direction some sort of geophysical process causes it to move around the circle in a fairly periodic way. However, we do not see that in the simulated data because there is no drift term like there appears to be in some cases in the data. This geophysical process is likely also driving the autocorrelation we see in the data. One possible extension of this model to better match the data could involve a drift term on the unwrapped angular data, such as in a wrapped ARMA process as described in [11]. Mixture models have also been used successfully, for example in [1] and several others. In any case, this model appears to be too simple for high-fidelity modeling of autocorrelated angular data with a given distribution. Another possibility is to employ dynamic harmonic regression, such as in the CAPTAIN toolbox for MATLAB ([25]), to model the several periodic components (daily, seasonal, etc) along with the drift to better match the data.



(a) Simulated data

(b) Real data

Figure 6.7: Unwrapped simulated and real data

7 Conclusions

We introduced the concept of directional or circular statistics and two commonly used distributions on the circle, and analyzed two simple methods of generating data with a von Mises distribution. We showed that the von Mises process shown in equation 5.7 can be used to generate an autocorrelated time series of angular data that has a stationary von Mises distribution, and we compared results from simulations using that process to wind direction data collected by the National Data Buoy Center in the Atlantic Ocean. While the model was able to roughly capture the autocorrelation of the data and would, given enough time, match the distribution of the deat as well, it required a large number of samples to accurately model the stationary distribution and did not match other properties of the data properly. We suggest that, while this model performs reasonably well given that it is exceedingly simple, it is likely more appropriate to simulate this sort of data using higher order models such as one of the many flavors of ARMA(p,q) models suggested in, for example, [11], among others. Several other such models are listed in the bibliography.

References

[1] Al Yammahi, Aishah & Prashanth R. Marpu & Taha B. M. J. Ouarda (2021), ”Modeling directional distributions of wind data in the United Arab Emirates at different elevations”, Arabian Journal of Geosciences

[2] Artes, Rinaldo & Clelia M. C. Toloi (2009) An Autoregressive Model for Time Series of Circular Data, Communications in Statistics - Theory and Methods, 39:1, 186-194, DOI: 10.1080/03610920802650338

[3] Berens, Phillipp (2009) CircStat: A MATLAB Toolbox for Circular Statistics. Journal of Statistical Software Volume 31 Issue 10

[4] Borradaile, G. J. (2003). Statistics of earth science data : their distribution in time, space, and orientation. Springer. ISBN 978-3-662-05223-5

[5] Broemeling, Lyle D. (1 November 2011). ”An Account of Early Statistical Inference in Arab Cryptology”. The American Statistician. 65 (4): 255–257. doi:10.1198/tas.2011.10191

[6] Coles, S. G. (1998). Inference for circular distributions and processes. Statistical Computing, 5

[7] Craig, Peter S (1988), ”Time Series Analysis for Directional Data”, Phd Thesis

[8] Fisher, N.I and A.J. Lee (1994), ”Time Series Analysis of Circular Data”, Journal of the Royal Statistical Society. Series B

[9] García-Portugués, Eduardo, Michael Sørensen, Kanti V. Mardia, and Thomas Hamelryck (2019), ”Langevin diffusions on the torus: estimation and applications”, Statistics and Computing

[10] Gubner, John A. (2006). Probability and Random Processes for Electrical and Computer Engineers. Cambridge University Press. ISBN 978-0-521-86470-1

[11] Harvey, Andrew; Hurn,Stan, and Stephen Thiele (2019) ”Modeling Directional (Circular) Time Series”, https://doi.org/10.17863/CAM.43915

[12] Holzmann H, Munk A, Suster M, Zucchini W (2006) Hidden Markov models for circular and linear–circular time series. Environ Ecol Stat 13:325–347

[13] Johnson RA, Wehrly TE (1979) Bivariate models for dependence of angular observations and a related Markov process. Biometrika 66:255–2566

[14] Kato,Shogo (2010); "A Markov Process for Circular Data" , Journal of the Royal Statistical Society. Series B (Statistical Methodology)

[15] Kent, J. (1975). Discussion of paper by K. V. Mardia. J. Roy. Statist. Soc. Ser. B, 37(3):377–378

[16] Kuiper, N. H. (1960) "Tests concerning random points on a circle". Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A. 63: 38–47.

[17] Larsen, Richard J and Morris L. Marx, "An Introduction to Mathematical Statistics and Its Applications, 6th Edition", Pearson Education, 2018

[18] Mahrt, Larry (2011) "Surface Wind Direction Variability", Journal of Applied Meteorology and Climatology,

[19] Mardia, Kantil; Jupp, Peter E. (1999). Directional Statistics. Wiley. ISBN 978-0-471-95333-3.

[20] Mardia, K. V. (1975), "Statistics of Directional Data", Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, March 19th, 1975

[21] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953), Equation of state calculation by fast computing machines, J. Chem. Phys., 21, 1087-1092

[22] National Data Buoy Center, https://www.ndbc.noaa.gov/

[23] Øksendal, Bernt (2007), Stochastic Differential Equations: An Introduction with Applications, Sixth Edition, Springer-Verlag Heidelberg New York

[24] Programmatic Environmental Assessment For the National Oceanic and Atmospheric Administration National Data Buoy Center (2018)

[25] Young, Peter C., Wlodzimierz Tych, C. James Taylor, The Captain Toolbox for Matlab, IFAC Proceedings Volumes, Volume 42, Issue 10, 2009, Pages 758-763

A Project git repository

The code and some of the references used for this project (the ones that were available online), along with other papers that were not referenced but that the reader may find interesting, are located in a github repository at https://github.com/mfleduc/Project-Circular-Statistics. To run the code, first run the script startAnalysis.m. The model is run in analysis/sdemodel.m. Other scripts in the analysis folder are mostly from starting or stopping down a certain path during modeling. Documentation is incomplete but at this point the code as written generally assumes angles are in radians.

The MATLAB code for the CircStat toolbox can be found on the MATLAB file exchange at https://www.mathworks.com/matlabcentral/fileexchange/10676-circular-statistics-toolbox-directional-statistics.