ECMI Modelling Week 2018
University of Novi Sad, Faculty of Sciences

WIND POWER PREDICTION

Tihomir Ivanov (Sofia University "St. Kliment Ohridski")

Miguel Ribeiro (University of Lisbon)
Johannes Stojanow (Technical University of Dresden)
Michi Rakotonarivo (University of Grenoble Alpes and Grenoble INP)
Megan Smith (University of Oxford)
Olof Ekborg (Chalmers University of Technology)
Victoria Heinz (Technical University of Darmstadt)

December 8, 2018

Contents

1 Introduction

Wind power as a renewable energy source is a growing factor in global electrical power supply. The share of wind power in the European Union's total installed power capacity has increased from 6% in 2005 to 18% in 2017 (in "Wind in power 2017-Annual combined onshore and offshore wind energy statistics") with support by the politics. However, the electrical power supplement system is designed for energy generation on demand, whereas the wind power generation is weather dependent. Therefore, the power supplement system is a combination from renewable energy generation and fossil fuels. In order to combine them in a smart and efficient way wind speed prediction is necessary.

For different applications, predictions in different time scales are utilized. A short time scale (48-72h) is important for energy trading and power supplement management, a longer time scale is used for maintenance. This report aims to predict the wind power in Sofia for six hours ahead. Therefore data from the weather station at the airport in Sofia from 01.06.2017 till 01.06.2018 for every half an hour is provided and includes temperature, pressure, wind speed and direction and meteorological information. Additionally, data from the Global Forecast System (GFS) is used. GFS is a weather forecast model produced by the National Centers for Environmental Prediction (NCEP)(*Global Forecast System*). First of all, we introduce the power curve to figure out the dependency of the wind power from the wind speed. Thus, only the prediction of wind speed remains. To do so, we apply two different approaches: the physical and statistical approach. In the statistical approach, the wind speed observations are considered as a time series and the distribution is investigated. Furthermore, the auto-regressive integrated moving average (ARIMA) model is introduced and trained with the provided data in order to determine suitable parameter for the model.
The physical approach starts from an atmospheric model based on the Navier-Stokes-Equations. We propose simplifications, which will lead to two methods. One using modified streamfunction-vorticity reformulation and the other a two-dimensional Burgers equation. Then both will be discretized in time and space with a finite difference approximation.

2 The Power Curve

The power curve connects the wind speed forecast with the desired wind power forecast. In order to calculate the power curve, data from wind speed with the corresponding generated wind power is utilized. The data includes measurements from one month in a time interval of one minute.
In Figure 1, the scatter diagram of wind speed with the corresponding wind power is depicted. Also three fitted piece-wise polynomials of the degrees 3, 4 and 5 are shown. Their coefficients are determined by the data. The polynomial function of degree 3 has a root-mean-square-error (RMSE) of 100.700, the polynomial function of degree 4 has an RMSE of 79.772 and the polynomial function of degree 5 has an RMSE of

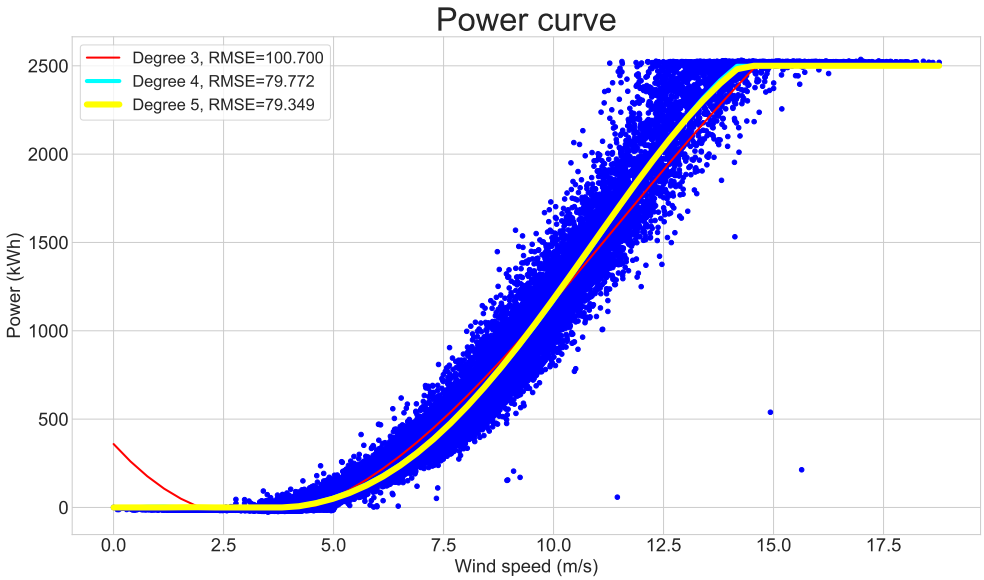


Figure 1: Historical data of wind speeds and the corresponding powers, as well as power curves fitted by the data as polynomial functions of various degrees.

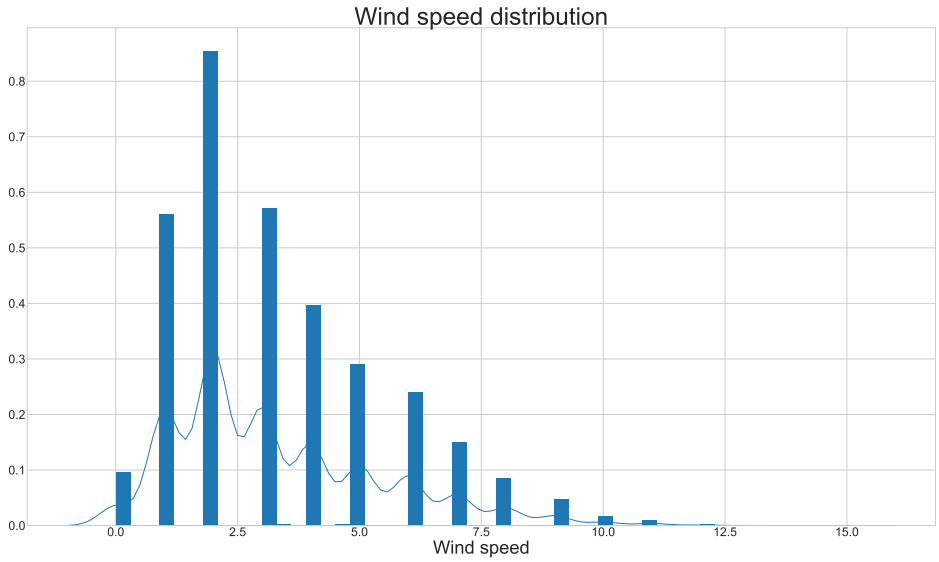
79.349. So the power curve fits the data sufficiently well and will be used to calculate the wind power. Therefore, the following presented techniques will concentrate on the wind speed forecast.

3 Wind Speed Prediction Methods

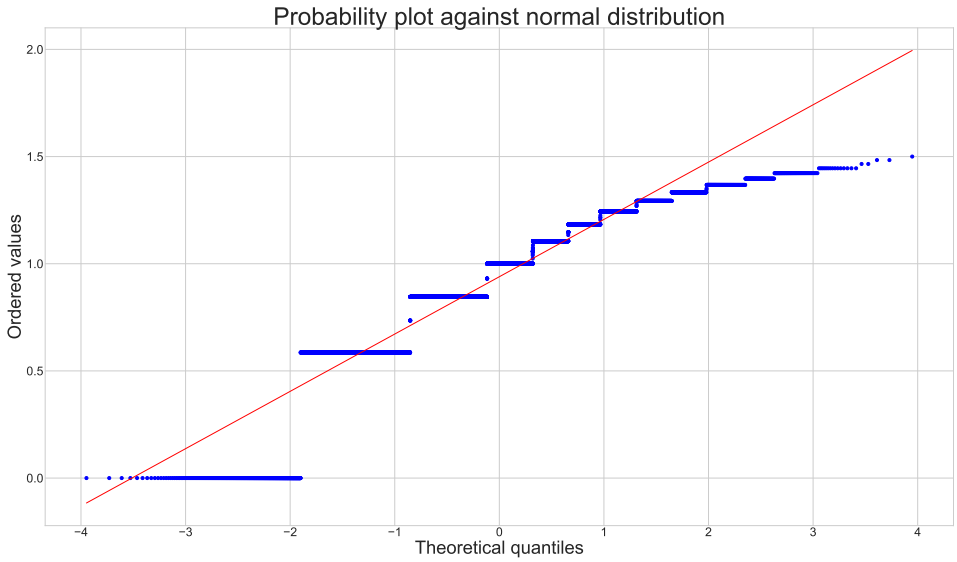
3.1 The Statistical Approach

3.1.1 Time series

Given the observations being sequentially ordered with respect to 30 minute separated time stamps, an instinctive initial approach is to consider the wind speed observations as a time series. However as many time series methods assume Gaussian distribution, a normality check to validate this assumption is necessary. The distribution of the considered wind speed is illustrated in Figure 2a, where a clear positive skewness is visible. The skewness is further illustrated in Figure 2b where the probability plot illustrates a poor fit to the Normal distribution.



(a) Wind speed distribution.



(b) Q-Q plot of wind speed observations against Gaussian distribution.

Figure 2: Validation of Gaussian distribution assumption.

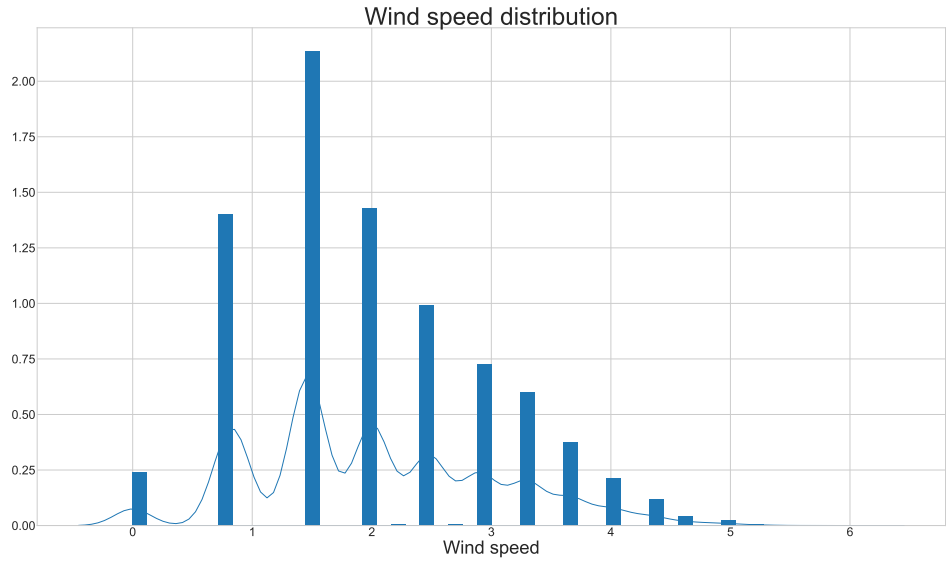
In order to obtain a Normal distribution of the observed wind speeds, a suitable power transformation is necessary. A common transformation for non-normal variables is the

transformation of Box and Cox 1964, which is defined as

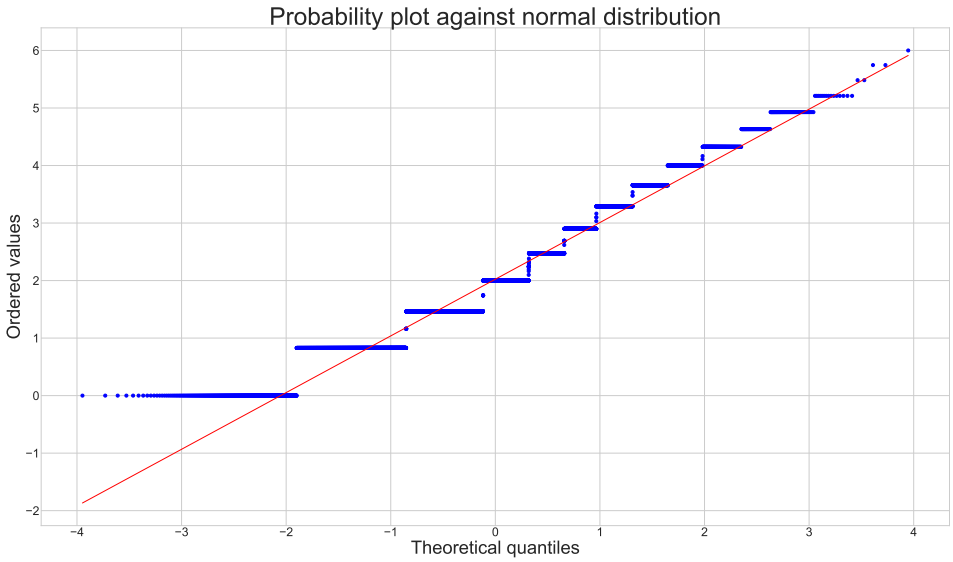
$$y_i^{(\lambda)} = \begin{cases} \dfrac{x_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln x_i & \text{if } \lambda = 0, \end{cases} \tag{1}$$

for some scale parameter λ. However as the Box-Cox transformation only accepts positive observations and the wind speed data contains several zero-values instances (as seen in Figure 2a), an alternative version of (1) is considered for $(x_i + 1)$ instead of x_i.

By thereafter incorporating a square root transform, i.e. $\lambda = 0.5$, the distribution will be given as illustrated in Figure 3a, where a slightly less skewed distribution is visible. The improvement is more obvious in Figure 3b where the probability shows alignment with the Normal distribution with a minor tail.



(a) Wind speed distribution after Box-Cox transformation.



(b) Q-Q plot of Box-Cox transformed wind speed observations against Gaussian distribution.

Figure 3: Validation of Gaussian distribution assumption after Box-Cox transformations with $\lambda = 0.5$.

In order to gain a more initiated understanding of the wind speed patterns, an additive deconstruction of the time series components is established such that

$$y_i = T_i + S_i + R_i \tag{2}$$

where T_i is the trend component, reflecting the long-term progression of the series, S_i represents the seasonal variation and R_i represents the residuals, i.e. noise of the ith observation. The decomposition of (2) for the last two weeks of observed wind speed in May 2018 is illustrated in Figure 4, where a very obvious daily seasonality is detected.

Since the seasonality will affect the wind speed value given time of its periodicity, the mean of the time series will be non-stationary and therefore unstable. In order to stabilize the time series from the daily seasonality, differencing is implemented. The differencing technique simply transforms each instance to be the difference between the current value and the value at the same periodicity of the previous season, i.e. $y_i = y_i - y_{i-48}$ since the daily seasonality consists of 48 half-hour intervals.

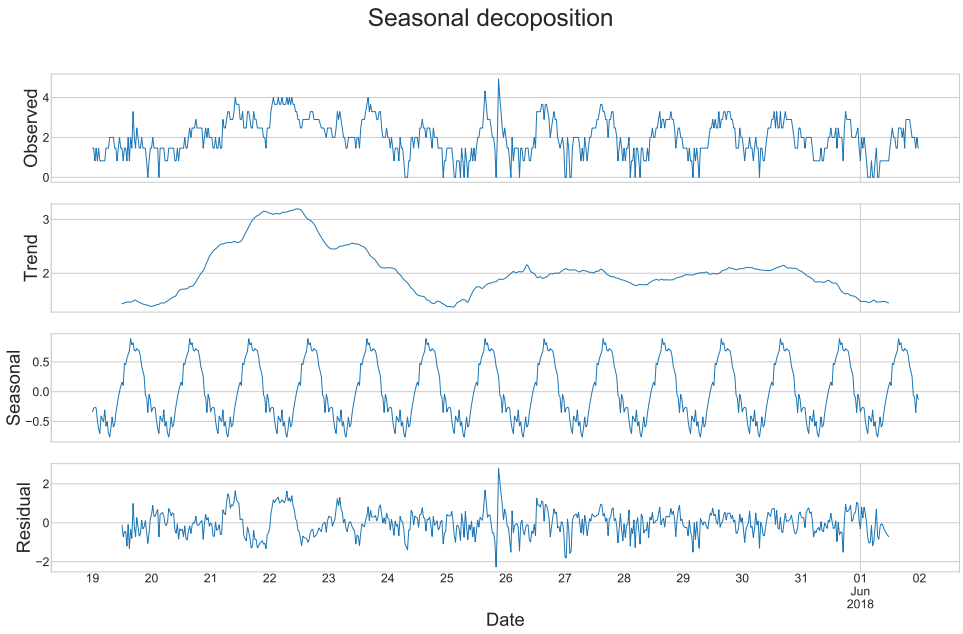
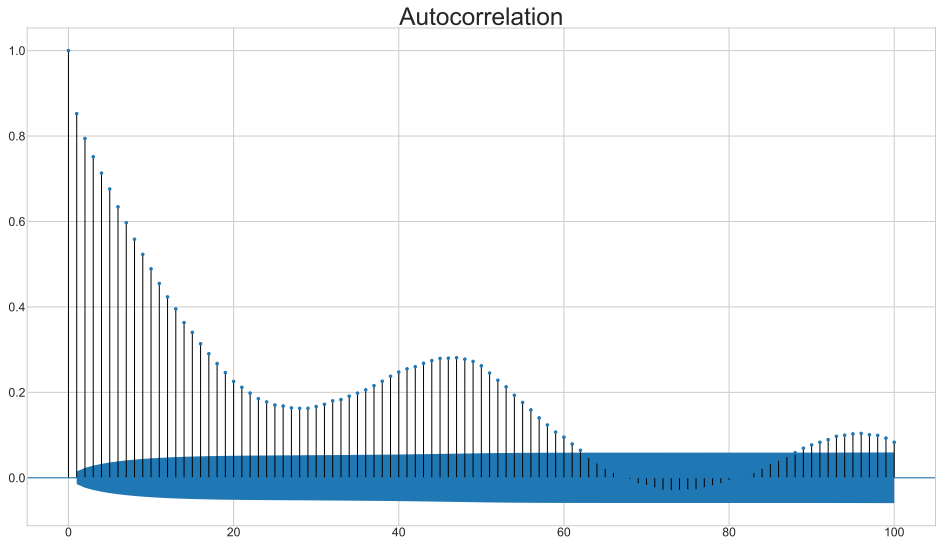


Figure 4: Seasonal decomposition of wind speed time series during last two weeks of May 2018.

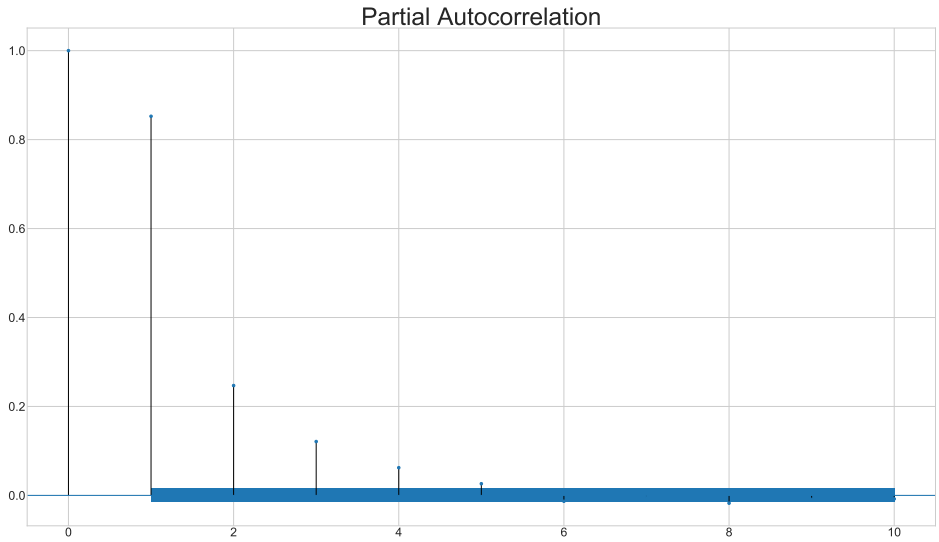
With non-stationarity being eliminated by differencing, an *autoregressive integrated moving average* (ARIMA) model from Box et al. 2015 is introduced. ARIMA is a generalized combination of three polynomials providing a greedy description of a stationary stochastic process. An ARIMA(p, d, q) model is fitted using an autoregressive part p, describing how the feature of interest is regressed on its lagged values, a moving average part q, describing how the regression error is a linear combination of error terms occurring at various past times, and a integrated part d, which indicates whether the data has been exploited to a differencing process.

In order to determined suitable parameter values for the ARIMA model, the correlations of the time series are analyzed. Figure 5 illustrates the autocorrelation, i.e. the observations correlation to lags, and partial autocorrelation, i.e. the autocorrelation with indirect correlations being removed, respectively. As seen in Figure 5b a cut-off at lag 6 is visible which, according to Box et al. 2015, indicates that $p = 6$ is a suitable fit for the data. Figure 5a however shows significant values for lags of order larger than 60,

indicating that the time series need to be integrated, i.e. $d = 1$, and that q could be rather large. For simplicity due to limited time however, $q = 1$ is chosen as an initial model estimation.



(a) Autocorrelation plot.



(b) Partial Autocorrelation plot.

Figure 5: Autocorrelation plots of Wind speed observations with a 95% confidence interval.

The initial model will hence be an ARIMA$(6, 1, 1)$ model. The model is initially trained on all observations prior to the last week of the time series, and a forecast of the next 4 hours is provided by the trained model. The test data will subsequently be added

to the training set and the model is retrained and tested on the next 4 hours of the test set. This iteration process will continue until the full test set has been considered, and the entire forecasting sequence of the last week of the time series is illustrated in Figure 6, with the total root mean squared error being 1.626. As seen in the figure, the model shows some delay in following the true drifts of the series, but overall the forecast captures the true series sufficiently well considering being an initial model.

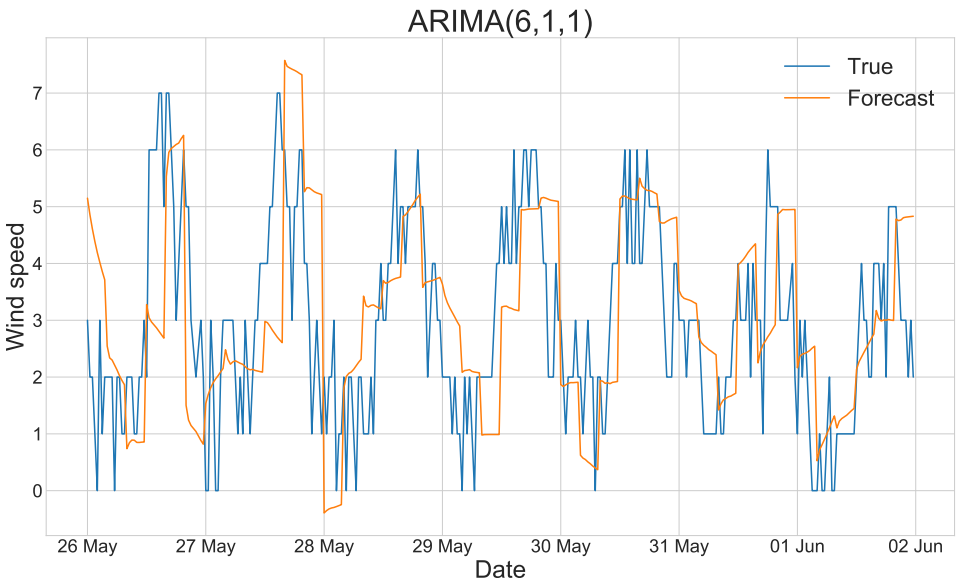


Figure 6: 4 hour wind speed forecast. Root mean square error of the forecast is 1.626.

3.2 The Physical Approach

As well as statistical methods, we also considered a more physical approach for predicting wind speed. Numerical Weather Prediction, commonly abbreviated to NWP, is practised regularly by weather forecasting companies to produce both short and long range forecasts. It achieves such results by generating a numerical simulation of the atmospheric governing equations, given some initial atmospheric conditions. The simulations predict the evolution of several meteorological quantities such as wind velocity, temperature, and pressure, and hence, it is possible to forecast wind speed in Sofia using this method.

However, the NWP method does not come without its flaws; such simulations require a high level of computing power, and the chaotic nature of the atmosphere introduces a reliance on the accuracy of the initial data provided. Nevertheless, the aim of this section is to demonstrate the capability and proficiency of NWP for wind speed prediction.

In the first subsection we introduce the full atmospheric model taken from Kalnay 2002, describe assumptions we made, and present the simplified model that we proceeded to use in our simulations. The following two subsections focus on two different methods that we implemented to predict how the wind velocity evolves in time using our initial data. The first method uses a modified streamfunction-vorticity reformulation to solve our simplified atmospheric model, whereas the second method simplifies our model further to the two-dimensional Burgers equation and produces results using a semi-implicit scheme.

3.2.1 Atmospheric Model

The governing physical equations of the atmosphere, often referred to as the primitive equations, are a set of coupled, non-linear, differential equations that are used to approximate atmospheric flow. They can be presented in several different forms but, since we are interested in predicting wind speeds in a small, localised area, we worked with the primitive equations on an f-plane, which is when the Coriolis parameter, f, is assumed to be constant. Taken from Kalnay 2002, these equations in Cartesian

coordinates are:

$$\frac{\partial u}{\partial t} + \bar{v} \cdot \nabla u = fv - \frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{3a}$$

$$\frac{\partial v}{\partial t} + \bar{v} \cdot \nabla v = -fu - \frac{1}{\rho}\frac{\partial p}{\partial y}, \tag{3b}$$

$$\frac{\partial w}{\partial t} + \bar{v} \cdot \nabla w = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g, \tag{3c}$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0, \tag{3d}$$

$$p = \rho R T, \tag{3e}$$

$$\frac{\partial s}{\partial t} + \bar{v} \cdot \nabla s = \frac{Q}{T}, \tag{3f}$$

$$s = C_p \ln \theta. \tag{3g}$$

In equations (3a) - (3g) $\bar{v} = (u, v, w)$ denotes the velocity vector, p represents pressure, ρ is density, T is temperature, s is the specific entropy of an air parcel, and θ is potential temperature. There are also some constants present in our system: f represents the Coriolis parameter, g is the gravitational acceleration, C_p is the coefficient of specific heat at constant pressure, and Q is the rate at which heat is supplied to an air parcel (Kalnay 2002).

System (3) is highly intricate and takes a lot of computational power to solve numerically. In order to demonstrate its capability in delivering wind speed forecasts, we simplify this system by assuming two-dimensional, incompressible fluid flow. This reduces system (3) to:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = fv - \frac{1}{\rho}\frac{\partial p}{\partial x}, \tag{4a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -fu - \frac{1}{\rho}\frac{\partial p}{\partial y}, \tag{4b}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{4c}$$

The next two subsections present different methods in which we can solve system (4).

3.2.2 Streamfunction-Vorticity Formulation

The Streamfunction-Vorticity formulation has often been used to successfully solve the two-dimensional incompressible Navier-Stokes equations using a finite difference algorithm (Salih 2013). As these equations are very similar to those that we wish to solve in system (4), we took the same approach. This approach involves reformulating system (4) using the vorticity vector, $\bar{\omega}$, defined generally as

$$\bar{\omega} = \nabla \times \bar{v},$$

and, more specifically in the two-dimensional $x - y$ plane, as

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \tag{5}$$

This reformulation also introduces the concept of a streamfunction, which is a scalar quantity defined for incompressible fluid flow in two dimensions. The fluid flow velocity components can be expressed as the derivatives of the streamfunction, Ψ, such that

$$u = \frac{\partial \Psi}{\partial y}, \quad \text{and} \quad v = -\frac{\partial \Psi}{\partial x}. \tag{6}$$

Equation (6) can be substituted into Equation (5) to obtain a Poisson equation for Ψ:

$$\nabla^2 \Psi = -\omega. \tag{7}$$

Equation (7) can be used in place of the continuity equation in (4c) since it is automatically satisfied by Equation (6). To complete the streamfunction-vorticity reformulation, another relationship between Ψ, ω, and the velocity components, must be obtained in addition to Equations (6) and (7).

The advantage of using the streamfunction-vorticity approach to solving System (4) is that it allows the admission of a solution without explicitly involving pressure. Differentiating Equation (4a) with respect to x and Equation (4b) with respect to y leads to a common mixed derivative of pressure in the two resulting equations. Subtracting one of these equations from the other eliminates pressure and results in the following vorticity transport equation:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = 0, \tag{8}$$

where we have simplified Equation (8) by using Equations (4c) and (5).

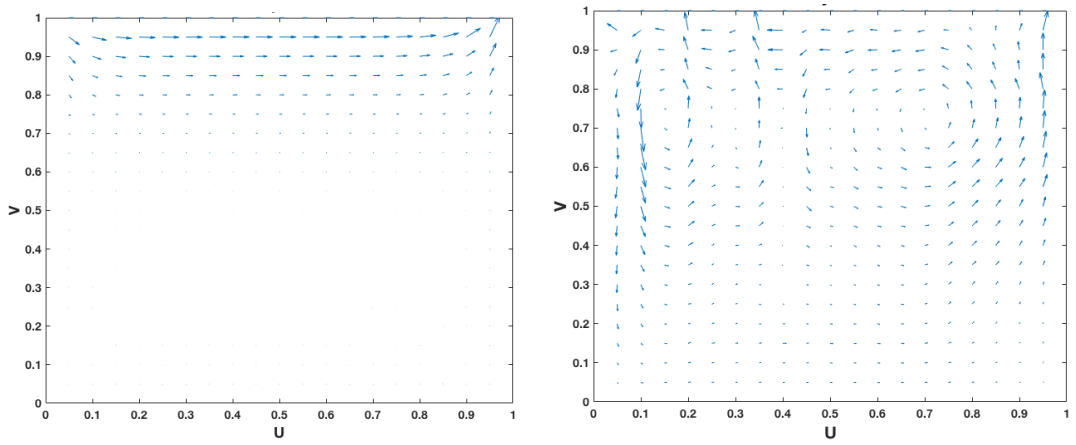
Together with Equation (6), Equations (7) and (8) provide all of the relationships necessary to be able to calculate the velocity components using ω and Ψ. By discretising all three equations, a finite difference algorithm can now be implemented to calculate how these quantities will evolve in time. However, we must first discretise the numerical spatial and temporal domains.

Let $D = [0,1] \times [0,1]$ be the spatial domain on which we wish to solve these equations. We begin by generating a uniform Cartesian grid over D with grid points (x_i, y_j) defined by

$$\begin{aligned} x_i &= 0 + i\Delta x, \qquad i = 0, 1, ..., N_x, \\ y_j &= 0 + j\Delta y, \qquad j = 0, 1, ..., N_y, \end{aligned} \tag{9}$$

where $\Delta x = \frac{1}{N_x}$ and $\Delta y = \frac{1}{N_y}$ represent the grid step sizes in the x and y directions, respectively and spacing measured by $h = \frac{1}{N+1}$. We also discretise the temporal domain by introducing time levels $t_n = t_0 + n\Delta t$, with a uniform time step $\Delta t = \frac{t_1 - t_0}{N_t}$. Here, N_t is the number of time steps that the simulation will take between the start time, t_0, and the final time, t_1. From here on, we use the notation $u^n_{i,j}$ to represent the value of variable u at grid point (x_i, j_j) at time t_n.

We can now begin discretising our equations. Since the Poisson equation in (7) is an elliptic partial differential equation, the standard central difference scheme is used to



(a) Wind velocity field at t_1=0.1 seconds. (b) Wind velocity field at t_1=2 seconds.

Figure 7: Evolution of the wind velocity field at different times in the simulation.

discretise the second order spatial derivatives, which gives

$$\frac{\Psi^{n+1}_{i+1,j} - 2\Psi^{n+1}_{i,j} + \Psi^{n+1}_{i-1,j}}{\Delta x^2} + \frac{\Psi^{n+1}_{i,j+1} - 2\Psi^{n+1}_{i,j} + \Psi^{n+1}_{i,j-1}}{\Delta y^2} = -\omega^{n+1}_{i,j}, \tag{10}$$

where Δx and Δy are the step sizes in the x and y directions, respectively. Equation (6) is approximated by using a first order one-sided finite difference scheme for both of its spatial derivatives, which means that we update the velocity components using

$$u^{n+1}_{i,j} = \frac{\Psi^{n+1}_{i,j+1} - \Psi^{n+1}_{i,j}}{\Delta y} \quad \text{and} \quad v^{n+1}_{i,j} = \frac{\Psi^{n+1}_{i,j} - \Psi^{n+1}_{i+1,j}}{\Delta x}. \tag{11}$$

For the vorticity transport equation in (8) we followed (Salih 2013) and used a third order upwind scheme for the discretisation of the first order spatial derivatives and the Euler forward difference scheme (otherwise known as the explicit Euler method) for the temporal derivative. Mathematically, this is written as

$$\begin{aligned} \frac{\omega^{n+1}_{i,j} - \omega^{n}_{i,j}}{\Delta t} + u^n_{i,j}\left(\frac{\omega^n_{i+1,j} - \omega^n_{i-1,j}}{2\Delta x}\right) + q(u^+\omega^-_x + u^-\omega^+_x) \\ + v^n_{i,j}\left(\frac{\omega^n_{i,j+1} - \omega^n_{i,j-1}}{2\Delta y}\right) + q(v^+\omega^-_y + v^-\omega^+_y) = 0, \end{aligned} \tag{12}$$

where $q = 0.5$ and

$$\begin{aligned} u^- &= \min(u^n_{i,j}), & u^+ &= \max(u^n_{i,j}), \\ v^- &= \min(v^n_{i,j}), & v^+ &= \max(v^n_{i,j}), \\ \omega^-_x &= \frac{\omega^n_{i-2,j} - 3\omega^n_{i-1,j} + 3\omega^n_{i,j} + \omega^n_{i+1,j}}{3\Delta x}, & \omega^+_x &= \frac{\omega^n_{i-1,j} - 3\omega^n_{i,j} + 3\omega^n_{i+1,j} + \omega^n_{i+2,j}}{3\Delta x}, \\ \omega^-_y &= \frac{\omega^n_{i,j-2} - 3\omega^n_{i,j-1} + 3\omega^n_{i,j} + \omega^n_{i,j+1}}{3\Delta y}, & \omega^+_y &= \frac{\omega^n_{i,j-1} - 3\omega^n_{i,j} + 3\omega^n_{i,j+1} + \omega^n_{i,j+2}}{3\Delta y}. \end{aligned}$$

Now that all of the key equations have been discretised, we are completely ready to implement the finite difference algorithm that will allow the prediction of the wind velocity components. The algorithmic process uses the following steps:

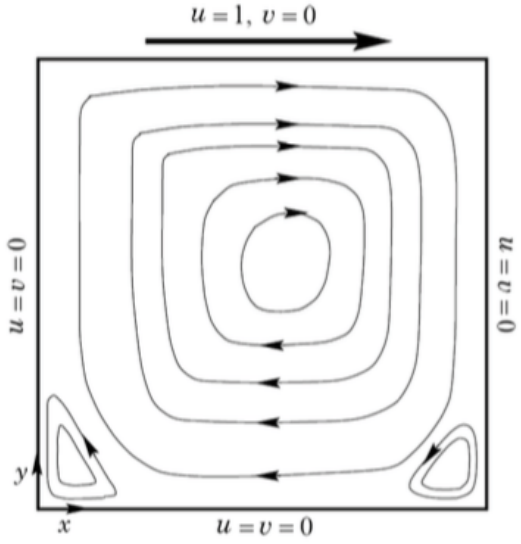


Figure 8: An illustration of the boundary conditions used in the simulation in Figure 7.

1. Use initial data for u and v to initialise ω and Ψ using Equations (10) and (11).
2. Compute ω at the next time using Equation (12).
3. Compute Ψ at the next time step by solving the linear algebraic system (10).
4. Compute u and v at the next time step using (11).
5. Return to step 2 and repeat procedure.

When we implemented this finite difference algorithm, we used initial wind speed data taken from Sofia to initialise the wind velocity field in step 1. We ran simulations for different values of t_1, the results of which are displayed in Figure 7, where we imposed the boundary conditions illustrated in Figure 8.

3.2.3 BTCS Configuration

A simple and intuitive approach is the direct substitution of partial derivatives with some low-order finite difference formulas. But first, we propose a further simplification of equations (4a), (4b) and (4c) by dropping the incompressibility condition (4c) and excluding all terms that contain the pressure p. This leads to the inviscid Burgers' equation in two dimensions

$$\frac{\partial u(x,y,t)}{\partial t} + u(x,y,t) \cdot \frac{\partial u(x,y,t)}{\partial x} + v(x,y,t) \cdot \frac{\partial u(x,y,t)}{\partial y} - f \cdot v(x,y,t) = 0, \quad (13)$$

$$\frac{\partial v(x,y,t)}{\partial t} + u(x,y,t) \cdot \frac{\partial v(x,y,t)}{\partial x} + v(x,y,t) \cdot \frac{\partial v(x,y,t)}{\partial y} + f \cdot u(x,y,t) = 0, \quad (14)$$

where f denotes the Coriolis force as a system parameter. Furthermore, we impose initial conditions and Dirichlet boundary conditions. Our domain is taken to be of rectangular shape, i.e. $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ with time line $[t_{\min}, t_{\max}]$. Next, we discretize the domain Ω and introduce

- $x_{\min} =: x_1 < \ldots < x_{N_x} := x_{\max}$ with equal width $h_x = (x_{\max} - x_{\min})(N_x - 1)^{-1}$,
- $y_{\min} =: y_1 < \ldots < y_{N_y} := y_{\max}$ with equal width $h_y = (y_{\max} - y_{\min})(N_y - 1)^{-1}$,
- $t_{\min} =: t_1 < \ldots < t_{N_t} := t_{\max}$ with equal width $\tau = (t_{\max} - t_{\min})(N_t - 1)^{-1}$.

Let $u_{i,j}^n$ and $v_{i,j}^n$ denote the numerical approximation to u and v at the grid point (x_i, y_j) at time $t = t_n$, respectively.

We follow the approach as in [Bahadir 2002] and discretize spatial derivatives with the well-known central difference stencil of second order. Since explicit time-stepping methods for transport phenomena lead to stability issues and therefore restrictions to the time step τ, we will use a simple first-order implicit method for the temporal derivative. This yields the following fully implicit "**B**ackward in **T**ime-**C**entral in **S**pace"-scheme:

$$0 = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau} + u_{i,j}^{n+1} \cdot \frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2 \cdot h_x} + v_{i,j}^{n+1} \cdot \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2 \cdot h_y} - f \cdot v_{i,j}^{n+1}, \tag{15}$$

$$0 = \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\tau} + u_{i,j}^{n+1} \cdot \frac{v_{i+1,j}^{n+1} - v_{i-1,j}^{n+1}}{2 \cdot h_x} + v_{i,j}^{n+1} \cdot \frac{v_{i,j+1}^{n+1} - v_{i,j-1}^{n+1}}{2 \cdot h_y} + f \cdot u_{i,j}^{n+1}. \tag{16}$$

We store the numerical solution at time step t_n as $W_n = (u_1^n, v_1^n, u_2^n, v_2^n, \ldots, u_N^n, v_N^n)$, where $N = N_x \cdot N_y$ and the single subscript index is according to the index map $\mathrm{ind}(i,j) = i + (j-1)N_x$, $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. To get the solution at t^{n+1} we need to solve the system $F(W_{n+1}) = 0$ of nonlinear equations, where $F \colon \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^{2 \cdot N}$ is a function containing the finite difference expressions for the solution, i.e. $F = (f_{u_1}, f_{v_1}, \ldots, f_{u_N}, f_{v_N})$. Fully expanding F, we have

$$F(W_{n+1}) = \begin{bmatrix} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau} + u_{i,j}^{n+1} \cdot \frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2 \cdot h_x} + v_{i,j}^{n+1} \cdot \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2 \cdot h_y} - f \cdot v_{i,j}^{n+1} \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\tau} + u_{i,j}^{n+1} \cdot \frac{v_{i+1,j}^{n+1} - v_{i-1,j}^{n+1}}{2 \cdot h_x} + v_{i,j}^{n+1} \cdot \frac{v_{i,j+1}^{n+1} - v_{i,j-1}^{n+1}}{2 \cdot h_y} + f \cdot u_{i,j}^{n+1} \end{bmatrix}_{i,j} \stackrel{!}{=} 0.$$

As in [Bahadir 2002], we apply Newton's Method to solve this system:

1. Choose an initial guess $W_{n+1}^{(0)}$, e.g. the previous time step solution $W_{n+1}^{(0)} = W_n$.
2. For $k = 0, 1, \ldots$ until convergence:
   - Solve the linear system $DF(W_{n+1}^{(k)}) \Delta W_{n+1}^{(k)} = -F(W_{n+1}^{(k)})$,
   - Set $W_{n+1}^{(k+1)} = W_{n+1}^{(k)} + \Delta W_{n+1}^{(k)}$.

To implement Newton's Method, we need the Jacobian $DF(W_{n+1}^{(k)}) \in \mathbb{R}^{2N_xN_y \times 2N_xN_y}$ of F or at least an approximation to it. In our case, we have the opportunity to write

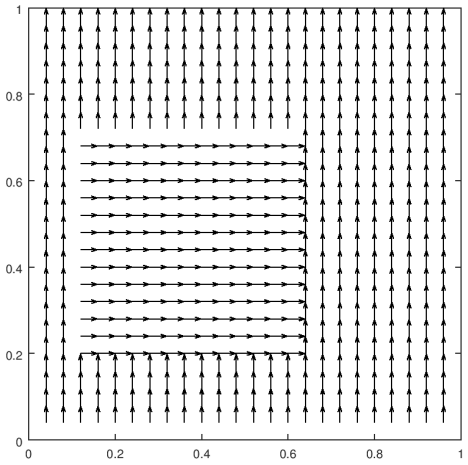
down an analytic expression for the Jacobian. Exploiting the linearity of F with respect to $u_{i,j}^{n+1}$, $v_{i,j}^{n+1}$, we get

$$DF(W_{n+1}^{(k)})_{i,j} = \begin{pmatrix} \frac{1}{\tau} + \frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2h_x} & \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2h_y} - f \\ \frac{v_{i+1,j}^{n+1} - v_{i-1,j}^{n+1}}{2h_y} + f & \frac{1}{\tau} + \frac{v_{i,j+1}^{n+1} - v_{i,j-1}^{n+1}}{2h_y} \end{pmatrix}.$$

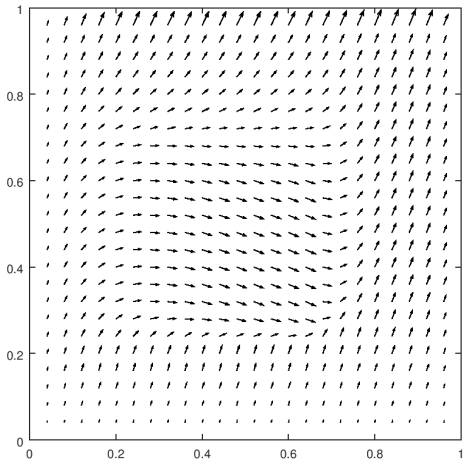
The full matrix is of size $N_x N_y \times N_x N_y$ and its entries are themselves block matrices of size 2×2. Note, that those finite difference expressions are only valid on interior nodes. On boundary index positions, we add identity block matrices with zeros on the right-hand-side. In doing so, we avoid undesirable corrections on the discrete boundary.

For illustrative purposes, we applied the BTCS scheme to the following test problem. Consider the system of Burgers' equations (13) and (14) on the unity square $\Omega = [0, 1]^2$. The parameter representing the Coriolis force is set to $f = 2$. The initial condition is a discontinuity in a sense, that for $(x, y) \in [0.1, 0.6] \times [0.2, 0.7]$, we have $u(x, y) = 1$ and $v(x, y) = 0$. Everywhere else, it is $u(x, y) = 0$ and $v(x, y) = 1$. Furthermore, we impose homogeneous Dirichlet boundary conditions. To see a quiver plot of the initial velocity profile, view Figure 9a. For the simulation, we set the time interval $T = [0, 2.5]$, the temporal step size $\tau = 0.1$ and the spatial step sizes $h_x = 0.04$ and $h_y = 0.04$. See Figure 9 for the evolution of the velocity field under a strong Coriolis effect. The Reynolds number is set to 100.

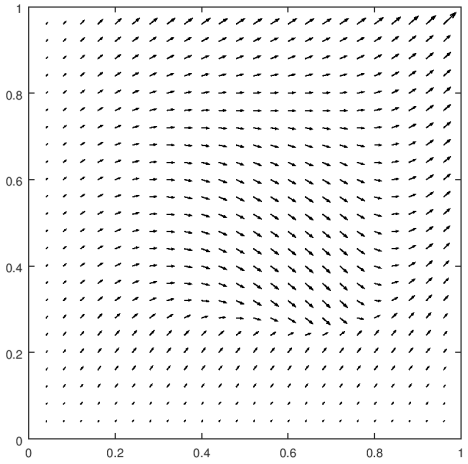
As the last point, we point out a direction, where this simple simulation strategy leads to. The computational domain will be chosen as a rectangle with its corners identical to nodes in the GFS database. To obtain as much accuracy as we can, we will choose the spatial resolution to be high. The temporal horizon of the simulation will be set to 4 hours as it is in the posed problem and the temporal resolution can be chosen quite low, due to the numerical method being implicit in time. This permits a big time step size, compared to explicit methods, which we don't use here. Furthermore, we will use pure Dirichlet boundary conditions, that will be incorporated from the Global Forecast System. Observations of wind speeds at the four corners of our domain will be taken from the GFS database and be interpolated at our grid nodes. Our boundary conditions will not evolve in time, but that will not bother us, as we are interested in one specific location only, namely Sofia Airport. The last adjustment will be the Coriolis force, which we then set to $f = 2 \cdot 7.2921 \cdot 10^{-5} \cdot \sin(42.6952)$.



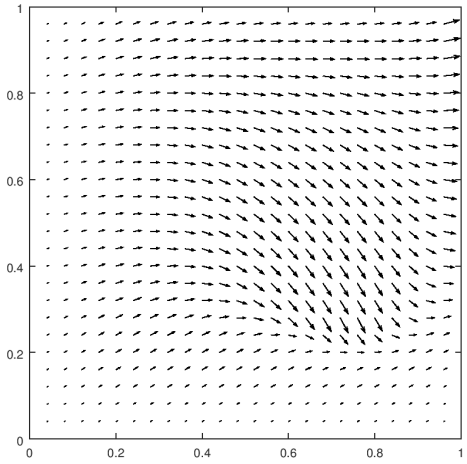
(a) Velocity field at $t = 0.0$



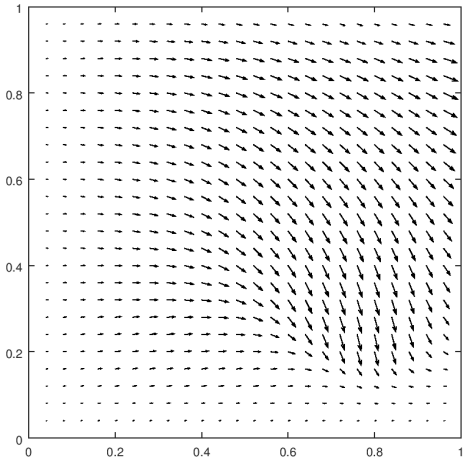
(b) Velocity field at $t = 0.5$



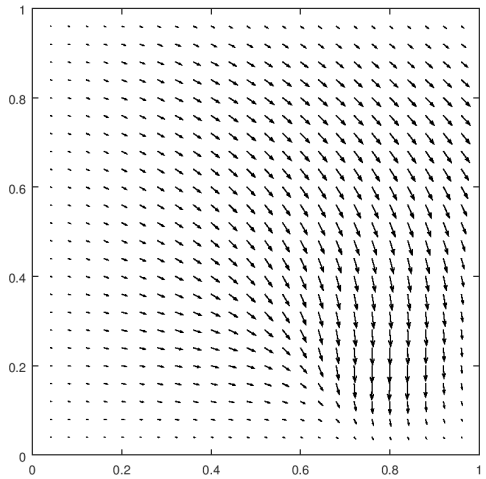
(c) Velocity field at $t = 1.0$



(d) Velocity field at $t = 1.5$



(e) Velocity field at $t = 2.0$



(f) Velocity field at $t = 2.5$

Figure 9: Evolution of the velocity profile during the simulation.

4 Discussion and Conclusion

First of all, the calculation of the desired wind power through a wind speed prediction using the power curve with satisfying accuracy is shown in Chapter 2. To achieve a wind speed prediction three methods are presented. In Chapter 3.1, the statistical approach with time series is described. Since the wind speed observations failed the normality check, a suitable power transformation is necessary. An auto-regressive integrated moving average (ARIMA) model is proposed and the parameters are fitted through analyzing the correlations of the time series. All in all, the wind speed forecast corresponds to true time series quite sufficiently, although it shows some delays. One way to improve the forecast is to find a more suitable model setting. In order to tune the parameters you can find a more accurate model with an extensive grid search. Besides more advanced methods could be implemented and compared in order find the most suitable method for the wind speed prediction.

In Chapter 3.2, the physical approach is specified. Although using two simplified methods based on the Navier-Stokes-Equations the implementation of the time and spatial derivatives is still challenging. Different finite difference approximations and their combinations are discussed to create a stable and robust algorithm. Both methods were tested with dummy data. Furthermore the provided real data should be included and the algorithm validated. Potentially the simplifications could lead to insufficient accuracy which should be investigated. Additionally, the algorithm could be analyzed regarding efficiency. In summary, the physical approach appears to be promising, but needs to be further evaluated due to the missing integration of the data. Subsequently, at this stage the physical approach cannot be compared to the other approach.

5 Group work dynamics

The problem of wind power prediction can be approached in a variety of ways, including numerous methods from both statistics and physics. With a variety of backgrounds and expertise within the Modelling Group, a decision was made to attempt a solution to this problem by implementing several of the different methods. In order for this to be achieved, the Modelling Group partitioned itself with the aim that each of its members would work on separate methods of solution and communicate their progress to each other throughout the Modelling Week.

This division of workload definitely has its advantages over simply focusing on one method of solution; more avenues were able to be explored, and it allowed for a direct comparison between the competence and efficiency of each method. However, it also comes with its disadvantages, which, unfortunately, were strongly felt in this instance. The division of group work also invoked a division in communication in the group. Segregated expertise afforded members to only have an understanding of the foundation and workings of their own method, rather than a shared knowledge of how all methods worked. Furthermore, with a large variety of backgrounds and expertise amongst the

members of the Modelling Group, an approach which encouraged a sharing of knowledge and closer group work activity may have been preferable.

This division of group work had a further knock-on effect when focus was turned to working together on this report. The job of relating all sections and aspects of this report together was difficult for a single member to convey leading to an informative but disjointed report. Moreover, in the instance that not all members could find time to contribute to this report, an entire section would have to be omitted from the report, due to the fact that no other members of the group understood it enough to explain it, which unfortunately was experienced here with one of the more practised methods from during the Modelling Week.

Whilst dividing the group into subgroups has its benefits, its drawbacks were more largely felt here. Successful work is only produced from a group that has been split up in such a way if a high level of communication can be maintained. In retrospect, choosing a different approach, wherein members of the Modelling Group work together on one method of solution may have delivered better results but, more importantly, it may have encouraged a Modelling Group that had better communication and a more energetic group dynamic.

6 Instructor's assessment

The problem of predicting wind speed for the purposes of wind power production is of high importance for the effective exploitation of the technology. There exist numerous approaches, both physical and statistical, and, therefore, the problem, posed during the ECMI Modelling Week, suggested many possible paths to be pursued.

The group worked effectively to outline a couple of possible approaches by doing some research and went on to the implementation of those ideas. From my point of view, the results, obtained during the Modelling Week were more than satisfactory, taking into account that for only a week the general trend of the wind behaviour was captured in the statistically-based forecasts. Even though the physically-based simulations include great simplifications of the atmospheric processes, they give a general idea of the possible paths to be further pursued.

Unfortunately, the results, obtained using artificial neural networks during the Modelling Week, are not included by the group in this report, even though they looked quite promising, due to the inability of a group member to write his part of the report.

References

Bahadir, A.R. (2002). "A fully implicit finite-difference scheme for two-dimensional Burgers' equations". In: *Applied Mathematics and Computation* 137, pp. 131–137.

Box, G. E. P. and D. R. Cox (1964). "An Analysis of Transformations". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 26.2, pp. 211–252. ISSN: 00359246. URL: http://www.jstor.org/stable/2984418.

Box, G.E.P. et al. (2015). *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley. ISBN: 9781118674925. URL: https://books.google.se/books?id=rNt5CgAAQBAJ.

europe, wind. "Wind in power 2017-Annual combined onshore and offshore wind energy statistics". In: URL: https://windeurope.org/about-wind/statistics/.

Global Forecast System. URL: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs.

Kalnay, E. (2002). *Atmospheric Modeling, Data Assimilation, and Predictability*. Cambridge University Press.

Salih, A. (2013). *Streamfunction-Vorticity Formulation*. URL: https://www.iist.ac.in/sites/default/files/people/psi-omega.pdf.